SPRING 1980 VOL. 2, NO. 1 $2.50 USA/$2.90 CANADA

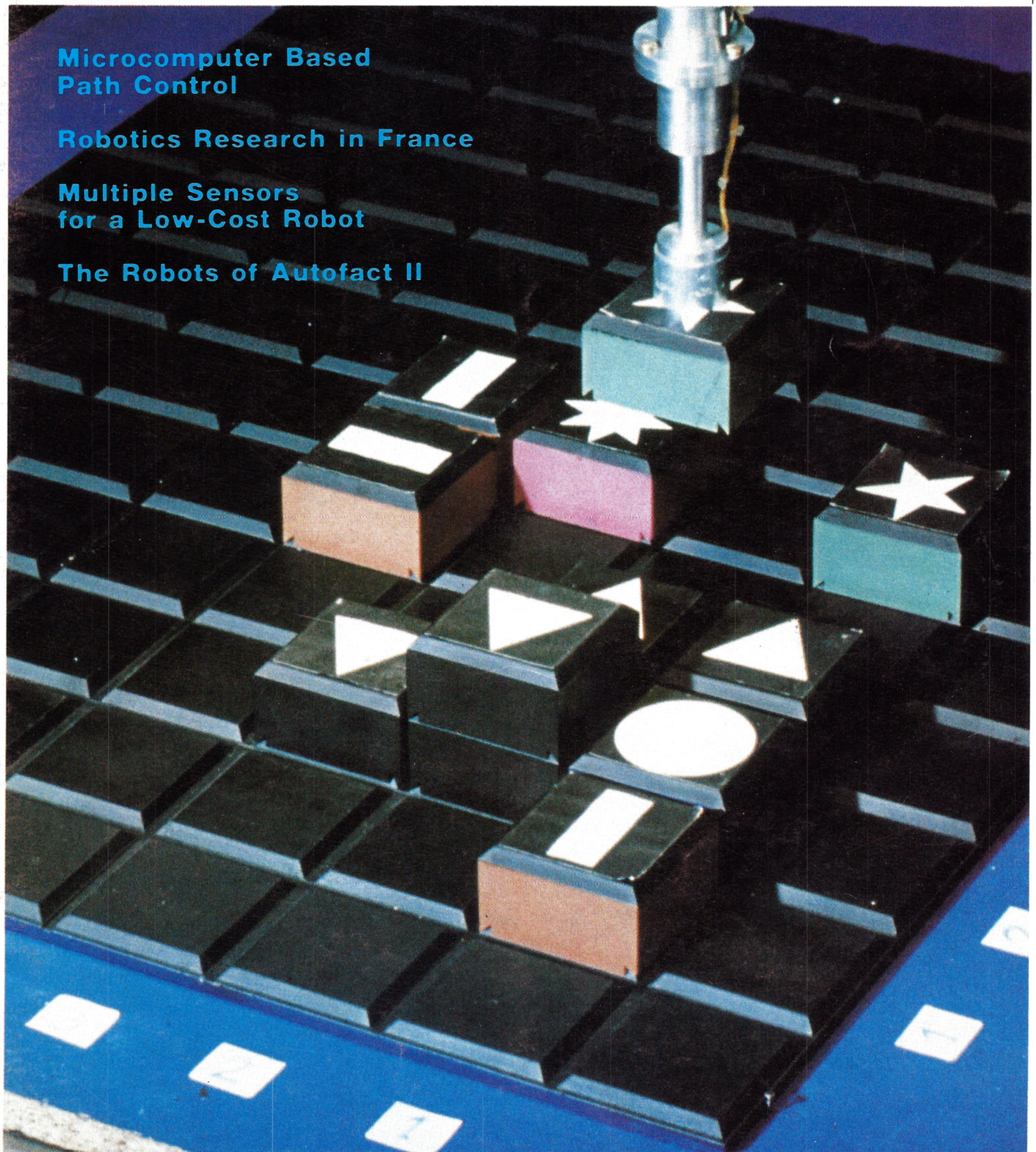
ROBOTICS AGE

Microcomputer Based Path Control

Robotics Research in France

Multiple Sensors for a Low-Cost Robot

The Robots of Autofact II



INDUSTRIAL ROBOTS

A Monthly Report on the Use of Robots in Manufacturing

VOLUME 1, NUMBER 1

If
you're worried. . .

. . . about the productivity of your manufacturing operation, it's about time you started learning more about industrial robots. And, a new publication—INDUSTRIAL ROBOTS—has been designed to help you do just that.

You're looking at the cover page of Vol. 1, No. 1. And, by filling out the coupon on this page you can get full details and the first issue, to appear in April, free.

Why a publication on industrial robots? Well look at productivity in the U.S. It virtually stood still in 1979 while industrial productivity in Japan has been moving up at better than 8% per year; England, 3%; and West Germany, 6%.

There are many reasons for this and industrial robots is one of them. The U.S. has just 3,000 industrial robots. Between Japan and Europe there are 6,000 working in plants and that number is growing rapidly.

Thus anyone trying to improve plant productivity must look at industrial robots as one way of accomplishing this. But as with any new field, this one is hard to keep track of.

There are meetings, journal articles, reports, patents—a constant stream of data spread over the world-wide manufacturing community. It would take a fulltime effort to sort out this mass of information, and pick out what is useful.

That's the job INDUSTRIAL ROBOTS will do for you. We'll be your eyes and ears for pertinent information on industrial robots. We'll attend technical meetings, read all those reports, check the patents, and most important, we'll tell you who is using industrial robots and for what.

Please send me full details on INDUSTRIAL ROBOTS and a free copy of the first issue . . .

Return this coupon to:
INDUSTRIAL ROBOTS
P.O. Box 1304
Fort Lee, New Jersey 07024

Name (please print) ______________________ Title ______________

Company ____

Address ____

City ____________________ State ____________ ZIP ____________

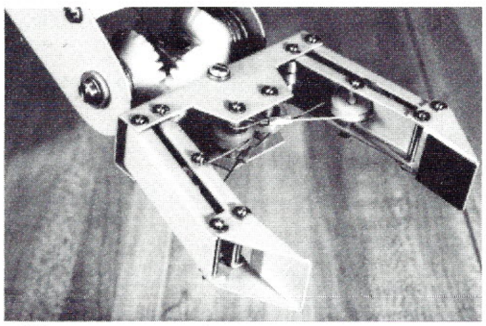
INDUSTRIAL ROBOTS is published monthly by **Technical Insights, Inc.,** 158 Linwood Plaza, PO Box 1304, Fort Lee, NJ 07024 USA. (Use PO Box address for all mail). One year subscription is $144, $12 additional for foreign airmail service.

Publisher: **Kenneth A. Kovaly.** Editor: **Dora K. Merris.** Managing Editor: **OyKue Brogna.** Contributing Editors: **Gail Henderson, Janet Olsen.** Asian correspondent: **Michael McAbee,** Tokyo. European correspondent: **Robin Johns,** London. Circulation Department—Director: **F. Kovaly;** Fulfillment: **Monica Gusk, Dorothy Baniel.**

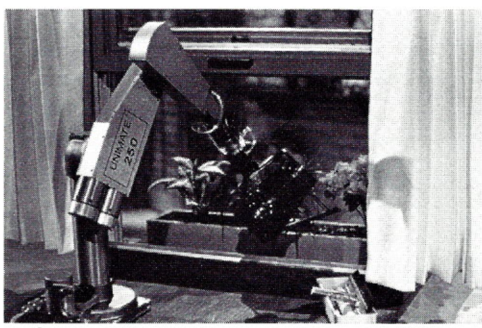
ROBOTICS AGE

SPRING 1980 VOL. 2, NO. 1

34



44



40

Articles

Departments

Publisher
Philip C. Flora

Editorial Director
Alan M. Thompson

Department Editor
Henry Fuhrmann

Editor
Hugh Bartel

Art Director
Robert Tinney

Production Editor
Cindee Huff

Technical Art
Heather Marr
James Marr

Artist
Andrew Layman

Advertising

Western States
Buckley & Boris
22136 Clarendon St.
Woodland Hills, CA 91367
213/999-5721

East, South & Midwest
Don Stockwell
International Assoc.
P. O. Box 4029
Houston, TX 77210
713/626-8436

ROBOTICS AGE is published four times a year, in the summer, fall, winter, and spring, by Robotics Publishing Corporation, 5147 Angeles Crest HWY, La Canada, CA 91011. Subscriptions are $8.50 for one year in the U.S.A. Application to mail at Second-Class Postage Rates is pending at La Canada, CA. **POSTMASTER:** Send address changes to ROBOTICS AGE, P.O. Box 801, La Canada, CA 91011.
Subscriptions from Canada and Mexico are $10 for one year. Other foreigh countries $15 for one year surface delivery. Single copy price is $2.50 in the U.S. and $2.90 in Canada and Mexico, $4 in Europe, and $4.50 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a U.S. bank. **Address all correspondence** to ROBOTICS AGE, P.O. Box 801, La Canada, CA 91011, phone 213/790-5823. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of ROBOTICS AGE.

LETTERS

He Likes Us!

Gentlemen:

I have received your Summer 1979 issue of ROBOTICS AGE Magazine and find it fantastic. Alan Thompson's article on robot vision is perhaps the best introduction to the field which I have ever seen. The article provides excellent coverage of a growing field. John Craig's DC motor article is also quite good. The New Products section is comprehensive and well put together. Overall, I must congratulate you on your extraordinary staff which puts out an extraordinary magazine. Please continue my subscription to this growing publication which will no doubt swiftly rise to the top of the field.

Tod Loofbourrow
Westfield, N.J.

Dear Tod,

Thanks for the compliments, and we intend to maintain the high standards established by our first issues. We want to congratulate you on the contribution to home robotics made by your new book (reviewed in ROBOTICS AGE, Vol. 1 No. 1). As mentioned in my editorial, (see page 5) we encourage our readers to start their own robot projects—we'll keep on supplying the necessary background and details.

—AMT

Treads for Traction

Dear Editor,

First of all, I wish to congratulate you on an excellent first issue of ROBOTICS AGE. I found the magazine to be a fine blend of theoretical and practical articles. Although I enjoy reading theoretical articles, I like having some circuits and programs which can be immediately used.

Now to one question which I hope you can answer. One form of motive power which I am interested in is tank tread. I am designing a rather large robot capable of carrying ni-cad batteries, 64K byte Z80 computer, voice I/O, two arms, camera and ultrasonic guidance. Since the robot must be capable of reaching at least 1.5m above the ground and handle fairly heavy objects up to 1m away, I require a broad, low slung base. For accurate tracking with such a heavy load, I consider treads a necessity.

Do you know of any companies which are likely to supply tread and/or tread units? I am looking for treads which are approximately 6cm wide and 1m in assembled length.

Raymond GA Cote, Consultant
PI Shoppe
Peterborough, NH

Dear Raymond,

Treads offer the advantage of spreading the load as well as the drive force over a large surface area, but they do so at the expense of increased power consumption (due to the forced skidding during a turn) and maintenance, as well as a more complex suspension and power train. Using minibike or motor-scooter wheels on a large wheelbase will give you all the stability you need, especially if you design the on-board weight distribution to balance the manipulator's payload. If you want to work outdoors and need better traction, consider using tires for an All Terrain Vehicle (ATV, such as Honda, etc.), which provide a large surface area with far fewer problems than treads. However, if you really want treads, try your local snowmobile shop.

—AMT

Due to unforseen delays, part II of the article Advances in Switched-Mode Power Conversion, *originally planned for this issue, was unavailable by our closing date. In our next issue, however, the exposition of the Ćuk switching converter will continue, along with practical circuit designs for robotics applications.*

One of the purposes of our Letters *department is to serve as a forum on robotics-related issues. Selected questions on robot design problems will be answered. We are always interested in hearing what subjects you would like to see covered in our articles. Send your comments, suggestions, and questions to:*

Editor
Robotics Age
PO Box 801
La Canada, CA 91011

EDITORIAL

You may not be rich and famous, but . . .

One reader wrote in saying that "the magazine is terrific. . .but give us hobbyists more designs we can use on our own low-budget robots." Subscribers who started with the second issue may have gotten the impression that we are placing too great an emphasis on professional and academic research, since no home-built circuits were featured. This would be misleading, however, since the articles in each edition of ROBOTICS AGE are contributed from various sources and will always vary in emphasis from issue to issue.

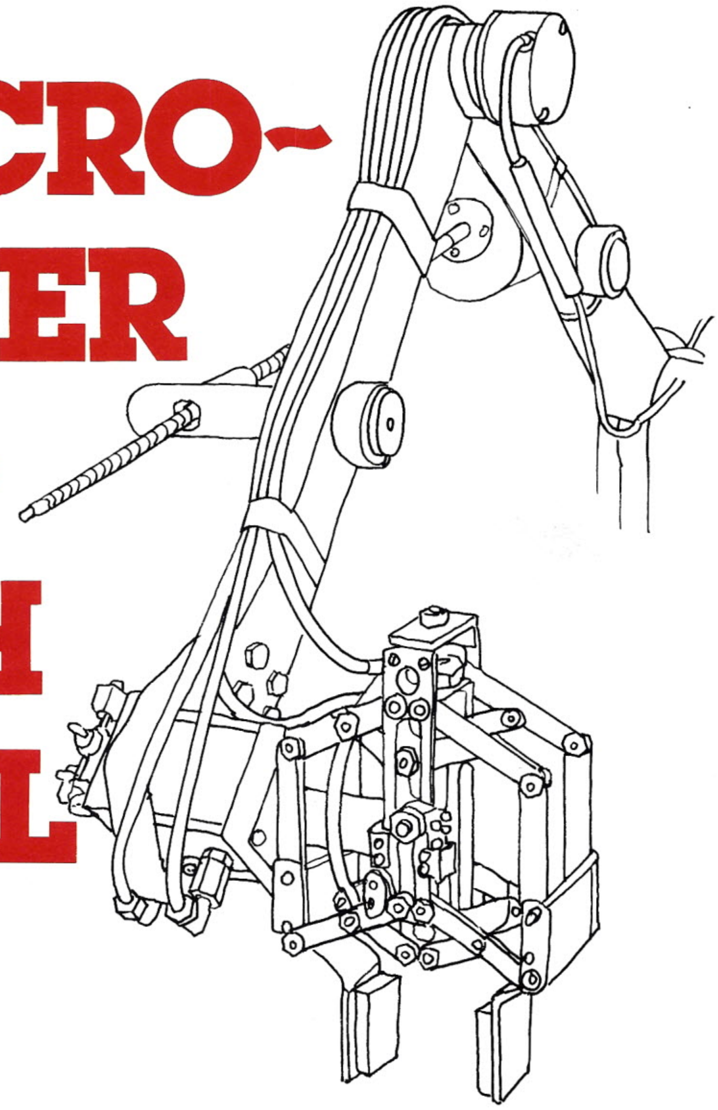
One of the basic goals of our editorial policy is to provide our readers with the knowledge they need to build their own robots, either for developing and experimenting with new ideas or strictly for education and entertainment. When we feature professional research, it is because of the importance of informing you of the latest developments in the field—and usually because these developments can, with a little imagination and effort, be adapted for use in low-cost systems. Just to take one example, the cable drive mechanisms used in the experimental robot manipulators at Japan's Electrotechnical Lab (described in Vol 1, No 2, Winter 1979) can be used in a variety of ways in home robots with many performance benefits.

We believe that the field of robot research is one in which any creative experimenter can make important contributions. Clearly, a formal background in control theory, mechanical design and artificial intelligence would be a great advantage, but to attempt to acquire expertise in each of these areas would take many years of study. The greatest contributions in any field have been made by those individuals who pursue their visions and acquire the specialized knowledge they need along the way. We know of many professionals who dismiss the work of amateurs, as well as many amateurs who regard academic research as impractical. Both, however, fail to appreciate the importance of pursuing both approaches—and this limited perspective is almost invariably due to chauvinism.

In our culture, the pursuit of scientific accomplishment or technological innovation has traditionally required that a person live and work in an academic environment and follow all the technical literature in the field. The advent of the personal computer has changed that requirement to a large extent. Now, with thousands working on the problem of integrating a microcomputer into a home robot, the real question is one of making the results of previous work, both professional and amateur, available to all. This is where ROBOTICS AGE comes in. But bear in mind that it is not *only* the professional robot researcher we are writing for—we intend to place a special emphasis on low-budget, microcomputer-controlled robots that any motivated experimenter can build. And we will help you learn all you need to know to understand both the theory and the technology leading to the intelligent robots of the future—and to start making your own contribution to robotics.

Alan M Thompson

MICRO-COMPUTER BASED PATH CONTROL



Prof. Wesley E. Snyder
Dept. of Electrical Engineering
North Carolina State University
Raleigh, N.C.

Introduction

The most frequent demonstration of industrial robots consists of a performance in which the robot is ordered to "Go to a point, stop, go to another point, stop, close the hand, go to another point, stop," etc. The robot is permitted to travel between stopping points along any path, usually unconstrained.

Programming a robot by specifying its task in terms of a series of points at which it must arrive and stop is not always possible. If the robot was using a paint sprayer, a sander or an arc welder, or carrying a tray of drinks, such start-stop motion would be totally unsatisfactory, if not disastrous.

For many applications, it is necessary for the robot's hand to move smoothly in space along a set of points which define its path (or "trajectory"). While point-to-point control can be implemented simply in hardware through relatively simple circuits, path control requires more complex algorithms which are generally most conveniently implemented in software. As we will show in this paper, an expensive computer is not necessarily required to implement this path control software. For many path control applications, a relatively cheap microprocessor is adequate. We will describe a system utilizing a slow eight-bit microprocessor, a Motorola M6800/D2 system, with a trivial amount of external hardware, which performs fairly

well.

There are two principal ways to specify a manipulator path: as a set of hand positions and orientations, or as a set of joint angles (see Figure 1). Which representation is most appropriate depends on the task.

In the most general of robot tasks, the robot must track and acquire some object moving in the robot's working area or apply a controlled force. To coordinate with the external world in this way requires the ability to control the hand in some fixed workspace reference frame. To move in this way, given the desired location in space, the computer must determine the proper torque to apply to each joint of the arm. Typically, this transformation from "Cartesian space to joint space" is a matrix operation in which the elements of the matrices are themselves trigonometric functions of the arm geometry. [1]

Fortunately, these extensive computations are not always necessary. Many applications require only that the robot follow a memorized path, such as the contours of an object to be painted. The operator may program the robot by manually guiding it through the desired path. A computer controlling each joint will remember the angles through which the joint moves (relative to some absolute zero reference point). These angles define the joint's position at every (discrete) point in time. Since this path may be memorized in terms of joint angles, real time computation of these coordinate transformations is consequently not required. In these cases, the path can be specified as a set of paths, one for each joint. The path for a joint is then simply a list of desired positions and corresponding times at which the joint should be in that position.

We will consider in this paper only the control of a single joint moving along a defined path. This implies using either several microprocessors, with one specifically assigned to each joint, or a time-division multiplexing system on a single processor. Both systems are in use today in different commercial systems.

A brief overview of the general control problem and servo design will precede a discussion of path control techniques which use these servos. We will discuss traditional (as distinguished from optimal) servo design, which is based on the concept of proportional error (PE) control to determine both the direction and magnitude of the torque to be applied to a joint. With traditional controllers come the traditional problems—steady state error and overshoot—and we will mention techniques for dealing with both of these.

We will then discuss the specific problem of path control and how one might design a suitable servo method for a path control system. We will describe two techniques, one based on control of position and one based on control of velocity.

The concepts described herein assume that the robot is operated by DC motors. We chose this context since it makes the ideas easier to explain. When one switches to another type of actuator, a hydraulic system, for example, the concepts are still valid, but some of the details must be changed.

Principles of Robot Manipulator Control

The basic principle of control is very straightforward: move the system in the direction that minimizes some error function. An example error function might be $E=\theta_d-\theta$, where θ_d is the desired angular position and θ is the actual position. (We will in future sections refer to θ_d as the "set point." When E=0, the joint is at the desired position. If E is negative, then the joint has moved too far and must reverse its motion. Thus, always moving in the direction which makes E appoach zero will provide a type of control.

Besides the drive direction, we should also be concerned with its magnitude. That is, not only must we ask, "In which way should I move the motor?," but also, "How much power (torque) should I apply to the motor?" Again, the error signal $E=\theta_d-\theta$ provides an answer. Let us apply a drive signal (a control) which is proportional to E. This rule

TIME (ms)	HAND POSITION & ORIENTATION						OR JOINT ANGLE					
	X	Y	Z	ROLL	PITCH	YAW	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
5	10	2	3	10°	20°	30°	10°	35°	70°	92°	18°	36°
10	12	2	3	11°	20°	31°	11°	34°	70°	90°	18°	36°
⋮												⋮

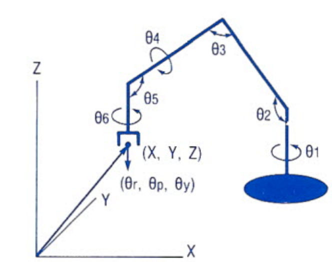


Figure 1. Two alternate ways of representing the path of a robot manipulator. This example assumes the robot arm has only angular joints. For a machine with linear joints, simply replace the appropriate θ_i by an "r_i."

defines a feedback control system as shown in Figure 2. Such a system is called a "proportional error (PE) control" system.

There are some problems with the proportional error control system which we have described. The first of these is the steady state error problem.

Consider the equation we have proposed for this servo: $T=K(\theta_d-\theta)$. That is, power applied to the motor (torque) is equal to some constant (K) multiplied by the error. Now, suppose we must hold a load against gravity. To do so requires a torque, so we cannot hold against gravity without an error, since no error would imply no torque. This is known as the steady state error problem.

A second problem with proportional error control is overshoot. That is, a manipulator operating under control of a proportional servo has only friction to slow it down. To see this, suppose the arm is close to its destination, then $T=K(\theta_d-\theta)$ is quite small, but not negative. If there is much friction, the arm may stop short of θ_d due to lack of drive, but if friction is small and inertia is large (relatively), then the arm may move on past θ_d. Now the error signal is negative, torque is backwards, and the arm will be driven back to θ_d. In the meanwhile, however, it has "overshot" its goal. If the task to be performed requires critical positioning, as in moving television picture tubes, for example, the occurrence of overshoot can be disastrous.

Let us now consider some techniques for dealing with steady state error and overshoot.

The Steady State Error Problem

One approach to dealing with steady state error is to output a torque $T=L+K(\theta_d-\theta)$ where L is a constant sufficient to hold the load when $\theta_d-\theta$ is zero. Use of this approach requires that the load be known precisely. In the case of robots, this knowledge is difficult to achieve since the load on a particular joint is usually a function of the positions and motions of the other joints.

An alternative is to make the drive signal equal to the integral of the error with respect to time. That is, allow the output of the servo (the motor torque) to accumulate with time, and make the rate of accumulation proportional to the error signal. Such a controller essentially *finds* the constant "L" defined above by the experimental technique of increasing L slowly until the load can be held stationary with no error.

A robot operating with "proportional integrating" (PI) controllers on its vertical axes can be observed to droop when a load is suddenly applied, and then rise back to the desired position.

Of course, the improved performance of an integrating controller does not come for free. We have been discussing how the integrator helps the steady state error problem. That is, integration is of assistance when the arm is stationary or moving slowly. When an integrating controller is used to achieve fast motion, however, it tends to increase the overshoot. In fact, under certain loading conditions, such controllers can be unstable and oscillate about the desired point. Thus, while reducing one problem, steady state error, we have made another problem, overshoot, even worse. Such are the joys of engineering (or government-controlled economies, for that matter)!

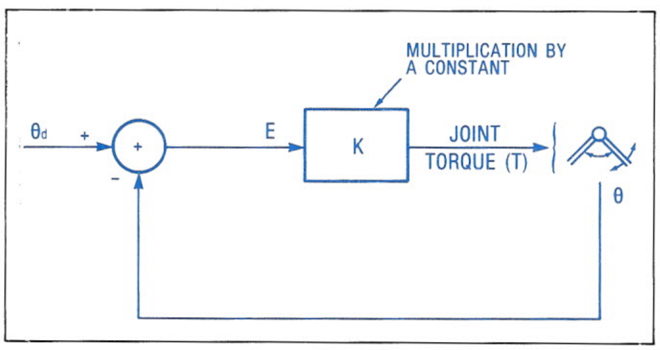


Figure 2. A Proportional Error (PE) control system, in which the control torque (T) is in proportion to the difference between the desired joint angle (θ_d) and its actual angle (θ).

The Overshoot Problem

One approach to this problem is to write differential equations which describe a PE or PI controller and its load (inertia, gravity, etc.). The solutions to these equations would then provide a mathematical insight into the conditions under which overshoot occurs. (Overshoot occurs when the solution is "underdamped.") However, in this paper we will pursue a more intuitive argument.

Overshoot occurs because the controller has an insufficient mechanism for "applying the brakes." A PE or PI controller in fact has nothing to stop the arm other than friction. If there is any positive error at all, $\theta_d-\theta$, is positive, and the motor will have positive (although small) drive applied right up to the point where the error goes to zero. If friction is small and inertia large relative to the friction, a joint driven by such a controller will overshoot.

To provide a degree of braking, we can use the following concept:

1) If error is large (we are a long way from the goal)

and the velocity is small, apply a large drive.
2) If error is small (we are close to the goal) and the velocity is high, apply a negative drive.

The simplest way to achieve this is to make the drive torque $T=K_1(\theta_d-\theta)-K_2\dot{\theta}$ where $\dot{\theta}$ is the angular velocity, the derivative (rate of change) of position with respect to time.

The ability of this controller to handle overshoot then depends on the *gains* of the controller, K_1 and K_2, and the inertia and friction of the load. One cannot always guarantee zero overshoot unless something is known about maximal values for inertia. Choice of optimal K_1 and K_2 is then possible. However, these constants are most often determined experimentally. Increasing K_2 is equivalent (for purposes of control) to increasing the friction of the system.

A controller with derivative feedback can then be combined with the concept of integration to yield a "proportional integral derivative" or PID controller. There are several ways in which one could configure such a controller. One such configuration is shown in Figure 3.

With a PID controller, one trades off the possibility of overshoot against the speed of the joint motion. Increasing K_2 tends to slow the arm down since it increases the negative contribution to torque due to velocity.

Because of their relatively simple implementation and "robustness" (ability to adapt to changing loads), PID controllers are probably the most commonly used controllers today, even though their performance is not optimal.

So far, we have discussed the servo implementation as if the total computation of the joint torque, T, was to be done in software. More and more robot manufacturers are going in this direction, although a variety of mixes of hardware and software are available. For example, one could close the servo loop in analog hardware such as operational amplifiers and use the microprocessor to adjust the gains for improved performance and to change the set point at the appropriate time. Alternatively, one could perform some of the servo computation in dedicated digital hardware.

Designing a Path Control System

Point-to-point control requires simply that the robot move to a specified point and stop, and then move on to the next point. To avoid the jerkiness provided by this type of control, we must develop a method to enable the robot to slide smoothly from one stopping point to the next along a specified path. Such path control can be accomplished in several ways by making use of the servo techniques discussed in the previous sections and imposing over those servos a control structure which dynamically modifies the set point.

In this section, we will discuss two techniques for achieving path control of a single joint. The first of these we call the "dog race" technique, in which the control computer simply moves the set point as the servo moves the joint towards the set point.* The second technique involves controlling the path by controlling the velocity along the segments of the path, thus insuring smooth transitions. We will provide a Motorola M6800 program listing of this method.

The Dog Race Technique

One way to achieve a continuous path is the "carrot and horse" or "dog race" technique. A computer tracks the moving robot and moves the goal before it is reached. Thus the robot, like the racing greyhound in pursuit of the mechanical rabbit, follows the path of the moving point.

However, the time required for the robot joint to move from one pont to another is affected by its interactions with other joints, its own inertia and friction and the inertia

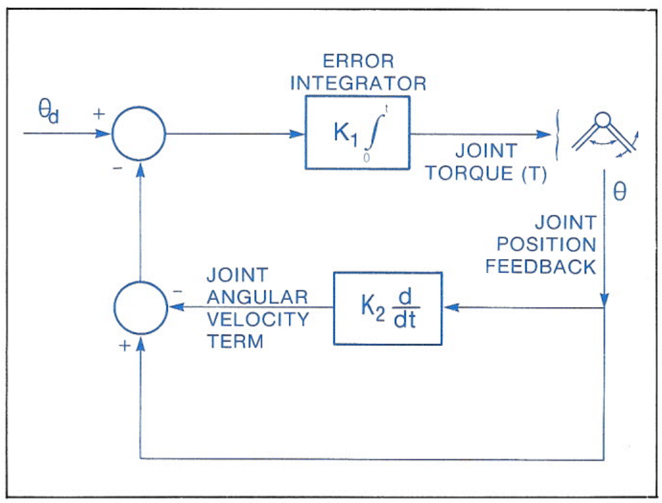


Figure 3. A Proportional Integral Derivative (PID) control system.

*Note: it is likely that the servo is simply software in the same computer as the path control algorithm. We distinguish between them only for pedagogical reasons.

of any load it might be moving. For example, if the robot is carrying a light load or if friction is less than was anticipated, the robot will tend to overshoot the goal (catch the rabbit) and stop. This can result in jerky start-stop movement. On the other hand, the friction may be extremely large, causing the robot to move more slowly. If a complicated path with forward and backward movement is designated, problems can develop since the robot may fall so far behind that it no longer moves along the desired path. (The greyhound may get so far behind that he finds the shortest distance to the rabbit is across the center field rather than along the rabbit's path on the track.).

There are numerous techniques for dealing with these problems. The most frequently used methods are:

1) Using a sampling rate (spacing between set points) which is appropriate for the robot in question. As a general rule, the time between set points should not exceed the mechanical time constant (response time) of the arm.
2) Making estimates of the load and adjusting the control gains (K_1 and K_2) appropriately. For example, one might compute loading at both ends of a trajectory and estimate loading along the trajectory by extrapolation.

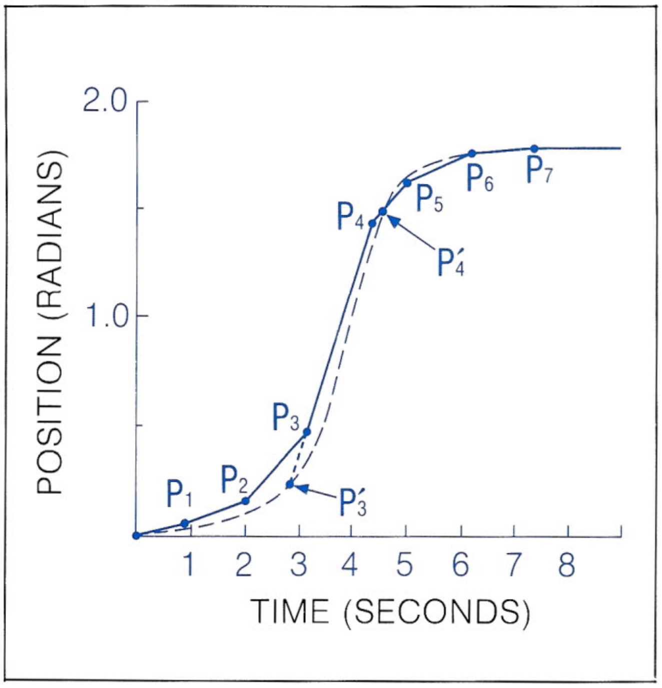


Figure 4. A typical trajectory for a manipulator joint, shown as a sequence of angular positions vs time. A possible joint response is shown by the broken line. The use of the intersection points P_3', P_4' will be explained later.

Quite good performance can be achieved by such systems when properly tuned.

A Technique based on Velocity Control

With the velocity control technique the time required for the robot to complete the path is divided into straight line segments. The robot is programmed to move at an assigned velocity which may increase or decrease at each time segment.

For example, Figure 4 shows a set of points (P_1, P_2, P_3...) at which the joint being controlled should be at particular times (T_1, T_2, T_3,...respectively). We have connected these set points with straight lines.

Each line segment has a constant slope. Since the slope is equal to the angular velocity, we can control the path by specifying the desired angular velocity at the appropriate time and then making use of a servo which controls velocity.

This method allows us to specify fewer points along the path than must be specified in the dog race technique and guarantees a reasonably smooth path.

There are some problems involved with correcting tracking errors in this method, and we shall deal with those in a subsequent section.

Since this method requires a servo which controls not position but velocity, some adjustments need to be made to the control algorithm given earlier. Thus, we will first consider how we might control velocity.

We will take the same PID controller developed in section 2 and apply it to control of velocity rather than position. In so doing, some computational simplifications result.

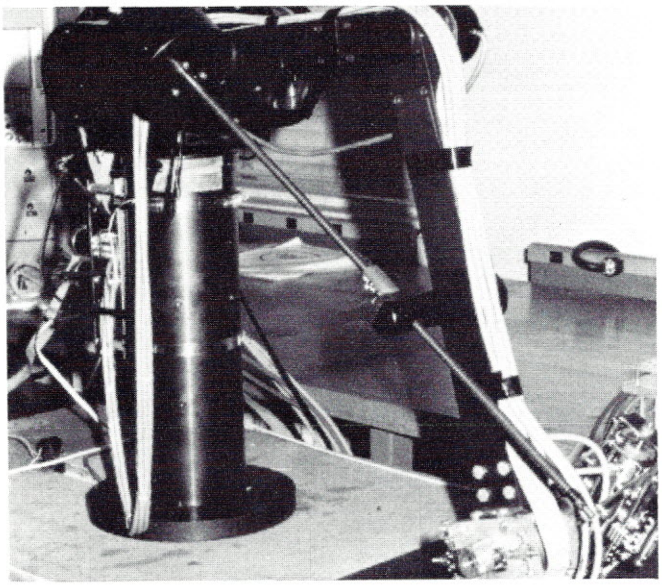
If we are controlling velocity rather than position, nothing about the structure of the PID controller must change, only the names of the variables and the quantities being measured.

The error signal e(t) is now equal to the difference between desired velocity and actual velocity:

$$e(t) = w_d - w(t)$$

where w(t) is the velocity at time t. The PID control signal is then:

$$k_i \int_0^t e(t')dt' + k_i k_d \int_0^t w(t')dt'$$



The microcomputer-controlled manipulator used in the NCSU experiments. Note the use of a lead-screw drive on the elbow joint. Apart from the arm itself, the system requires very little external circuitry, since most of the control functions are performed in software.

Note, however, that the term

$$k_i k_d \int_0^t \dot{w}(t')dt'$$

is obtained by first taking the derivative of the velocity feedback to find its rate of change $\dot{w}(t)$ (i.e., acceleration) and then integrating it. Since these two operations cancel one another, the same result can be obtained by using the velocity feedback in the control loop directly. Also, the process of differentiation tends to enhance noise and errors and should be avoided whenever possible. This can be accomplished by avoiding both the differentiation and integration, utilizing the control structure shown in Figure 5, where the new term $K_d' = k_i k_d$. This system implements a control method identical to that provided by the PID controller shown in Figure 3, but avoids the differentiation operator. Consequently it is called a "pseudo-derivative feedback" [2] controller.

Another simplification to the controller can be made by observing that the integral of error $e_I(t)$ is

$$e_I(t) = k_i \int_0^t e(t')dt' = k_i \int_0^t (w_d - w(t'))dt'$$

$$= k_i \int_0^t w_d(t')dt' - k_i \int_0^t w(t')dt'$$

but, since the integral of velocity gives the position, the term

$$\int_0^t w_d(t')dt'$$

is just the desired position (where the arm should be at any instant of time), and

$$\int_0^t w(t')dt'$$

is the arm's actual position, which can be measured directly by feedback. Thus, we can also avoid integration entirely by calculating the joint's positional error and multiplying it by k_i. The desired position can be obtained by solving a straight line equation of the path segment being servoed.

Consequently, we have implemented a "proportional, integrating, differentiating" controller without the necessity of either integrating or differentiating.

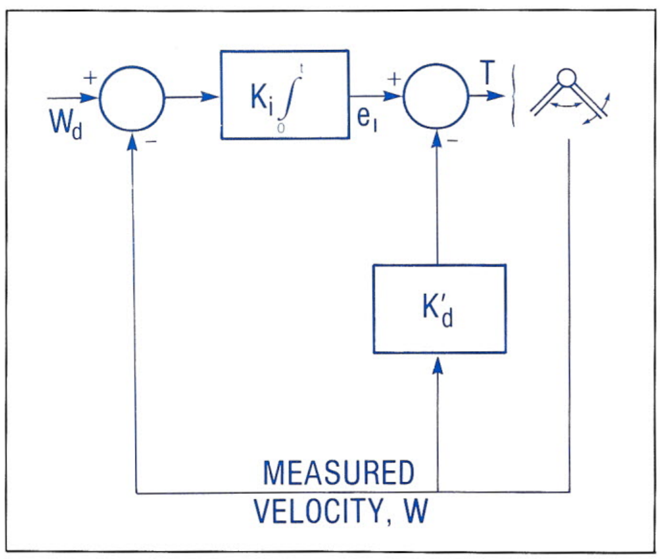


Figure 5. A Pseudo-Derivative Feedback (PDF) controller. Differentiation is avoided by using the feedback term directly, after the error integrator.

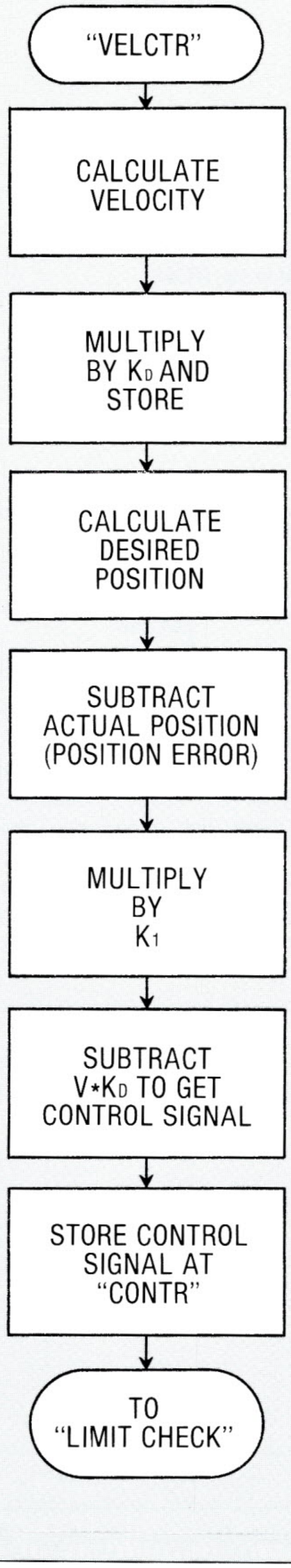


Figure 6. Flowchart of PDF velocity control signal calculations.

MOTOROLA 6800 PROGRAM

```
        * CALCULATION FOR PDF CONTROL SIGNAL
VELCTR  JSR       VELCY      READ VELOCITY
        JSR       DISPLAY    DISPLAY IT
        * MULTIPLICATION OF VELOCITY BY KD1
        * SINCE THE VELOCITY IS MEASURED AS
        * 1/16th OF THE ACTUAL VELOCITY, KD1=1/4
        * IS OBTAINED BY MULTIPLYING THE
        * VELOCITY WITH 4.
        * THE RESULT IS A 16 BIT BINARY NUMBER
CNTCAL  CLR  A               CONVERT THE
        LDA  B  VELTY        MEASURED VELOCITY TO
        BPL     K2           A 16 BIT NUMBER.
        COM  A
K2      LDX     #$2          MULTIPLICATION BY 4
K3      ASL  B               DONE BY 2 ROTATE
        ROL  A               LEFTS
        DEX
        BNE     K3
        STA  A  CONTR        SAVE KD1*VEL AT
        STA  B  CONTR+1      CONTR
        LDX     TPNTR
        * CALCULATION OF
        * INTEGRAL(VDESIRED-VACTUAL)*K1
        * WHERE K1 IS A CONSTANT
        * SINCE ABOVE IS EQUAL TO
        * K1*(DESIRED POSITION-ACTUAL POSITION)
        * WE CAN AVOID INTEGRATION BY CALCULATING
        * P. DESIRED AND SUBTRACTING P. ACTUAL
        * FROM IT.
        * CALCULATION OF K1*INTEGRAL(VDES-VACT)
        SEI
        LDA  A  CLOCK        GET X (TIME)
        LDA  B  CLOCK +1
        CLI
        SUB  B  1,X          PUT X-X1
        SBC  A  0,X          IN THE MULTIPLICAND
        STA  A  MULTND
        STA  B  MULTND+1
        LDA  A  4,X          PUT DESIRED VELCTY
        JSR     THETA        GET VEL*(X-X1)
        LDX     TPNTR
        LDA  A  2,X
        LDA  B  3,X          GET Y1
        * Y=Y1+VEL*(X-X1)
        * ASSUMING VEL*(X-X1) IS LESS THAN 32K
        * IN MAGNITUDE, THE RESULT WILL BE
        * IN LEAST SIGNIFICANT BYTES.
        ADD  B  RESULT+2     GET DESIRED
        ADC  A  RESULT+1     POSITION
        * CALCULATE INTEGRAL ERROR AS YDES-YACT
        SEI
        SUB  B  COUNT+1
        SBC  A  COUNT
        CLI
        * CALCULATION FOR VELOCITY ERROR.
        LDX     #2
INTDVD  ASL  B
        ROL  A
        DEX
        BNE     INTDVD
        * ASSUMING THAT THE POSITIONAL ERROR IS
        * ALWAYS LESS THAN 8K IN MAGNITUDE,
        * THE RESULT WILL BE IN LS BYTES.
        SUB  B  CONTR+1
        SBC  A  CONTR
        STA  A  CONTR
        STA  B  CONTR+1
        .
        .
        .
        (PERFORM LIMIT CHECKING)
```

Figure 7. PDF control method implemented in M6800 assembler code. Implementations for other microcomputers would use similar logic.

The flow chart for PDF velocity control is given in Figure 6, and a Motorola 6800 program is given in Figure 7.

Choosing the Change Point

In the last section, we proposed to implement path control by describing the path as a series of straight lines (regions of constant velocity).

This rather naive approach suffers from the fact that velocities cannot change instantaneously. Consequently, due to inertia, there is a time lag before the system achieves the desired velocity. Thus, there is always a positional error.

Figure 4 shows a desired joint path (solid line) for a typical motion, and the expected response (broken line).

One method to partially compensate for this lag is to command a new velocity earlier than specified. For example, rather than command velocity 4 at time P_3 in Figure 4, one might command it at time P_3', where the actual trajectory meets the extrapolated next segment.

We can determine when we cross the extrapolated next segment by the following procedure:

The equation of a line passing through a given point (x_1, y_1) and having a known slope "c" is

$$y = y_1 + c(x-x_1)$$

or

$$y-y_1-c(x-x_1) = 0$$

To determine whether a given point is above or below the given line, we substitute the x and y values for the point in the left hand side (LHS) of the equation above and test the sign of the result.

If LHS>0, the point is above the line,
If LHS=0, the point is on the line,
If LHS<0, the point is below the line.

To determine the point where the actual trajectory cuts the next segment, we first have to decide the region from which we are approaching the line. Once we know the region of our approach, to find the change point, we just look for the trajectory to move into the opposite region. The region of approach is determined by comparing the present desired velocity with the next desired velocity. This essentially tells us whether we are about to accelerate or decelerate. The two cases are:

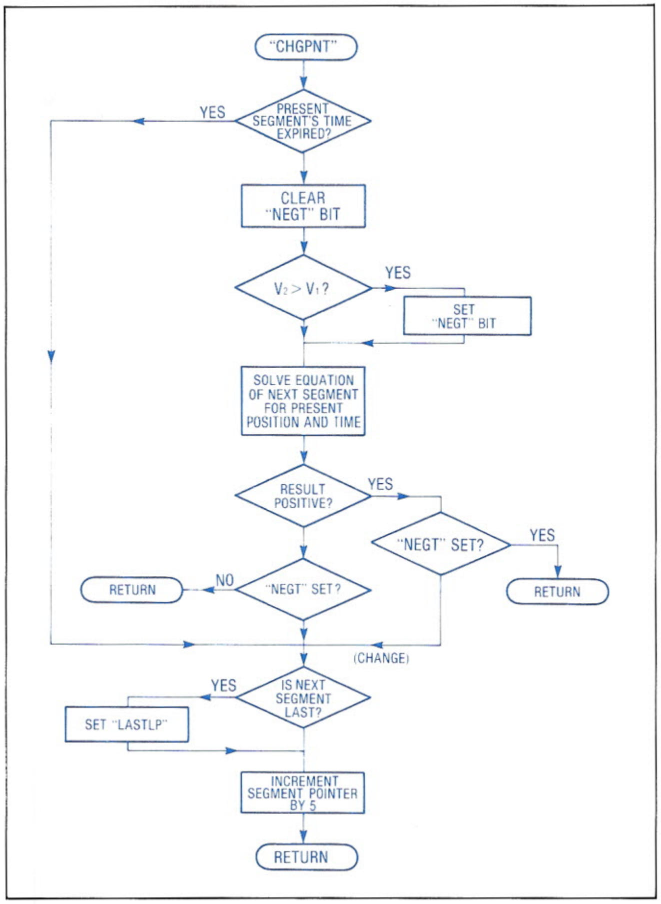


Figure 8. Flowchart of the program to find the change point between path segments.

1. If the next desired velocity is greater than the present desired velocity, we are approaching from a positive region, and should look for a negative value of LHS to change the desired velocity.
2. If the next desired velocity is smaller than the present desired velocity, we are approaching from a negative region and should look for a positive value of LHS to change the desired velocity.

There are some problems with the criteria given above for changing segments. In segment P_3—P_4, the joint acquires a very high velocity and if we wait until point P_4' to change velocities, we will be guaranteed to overshoot. A simple solution to this problem which yields satisfactory results is given by the following heuristic: Always change the desired velocity at the end of a segment, even if the actual velocity has not reached the next segment yet. On the other hand, if the actual trajectory reaches the next segment earlier than the "end of segment" time, the change is made as described earlier. Essentially, the heuristic says to always change at the earlier time, either when you cross the next segment or when the schedule specifies.

A flow chart of the algorithm for determining the change point is given in Figure 8.

Implementation and Performance

Our hypothesis was that one can do path control very inexpensively and simply. To test this hypothesis, we implemented this path control algorithm on a Motorola M6800/D2 system. This is an eight-bit microprocessor with no multiply or divide instructions, running at a reduced clock speed (650kHz). The entire program resided in less than 1k of ROM and required less than 512 bytes of RAM. The microprocessor board had no user communication features other than a hexadecimal keyboard/display.

Positional information was derived from a 2500 count/rev. incremental optical shaft encoder, which was connected to the computer to provide an interrupt once every ½ cycle of the encoder (approximately every 0.07 degrees). The computer incremented (or decremented) a memory location every interrupt and consequently kept track of position totally in software. Time information was also kept totally in software, derived from an oscillator chip which provided an interrupt every millisecond.

Velocity information was derived by some special purpose hardware which measured the width of the encoder pulses. This consisted of essentially a counter, a delay, and a latch. The total amount of off-board hardware used was eight TTL chips.

The total cost of the control system hardware was under $200, less than the cost of a single encoder.

The performance of the system is shown in Figures 9, 10, and 11. The desired trajectory is shown by the dotted lines with set points indicated. The actual path is given by the solid line.

Conclusions

In this paper, we have dealt with a small aspect of robot control, control of the path of a single joint.

Controlling the path of the robot as a whole can be decomposed into control of the paths of the individual joints. Thus, if we can control the paths of the individual joints as functions of position and time, we can control the path of the hand in space.

We have discussed two techniques for performing path control at the joint level. One technique based on velocity control has been developed in considerable detail. This technique has the advantages of extremely simple implementation even with a very limited hardware configuration, and reasonably good performance.

We have tried to show that path control is achievable with simple hardware and minimal software. We hasten to point out however, that this technique, at least in the form described here, is far from the solution to the world's problems. In particular, it requires the path to be known in

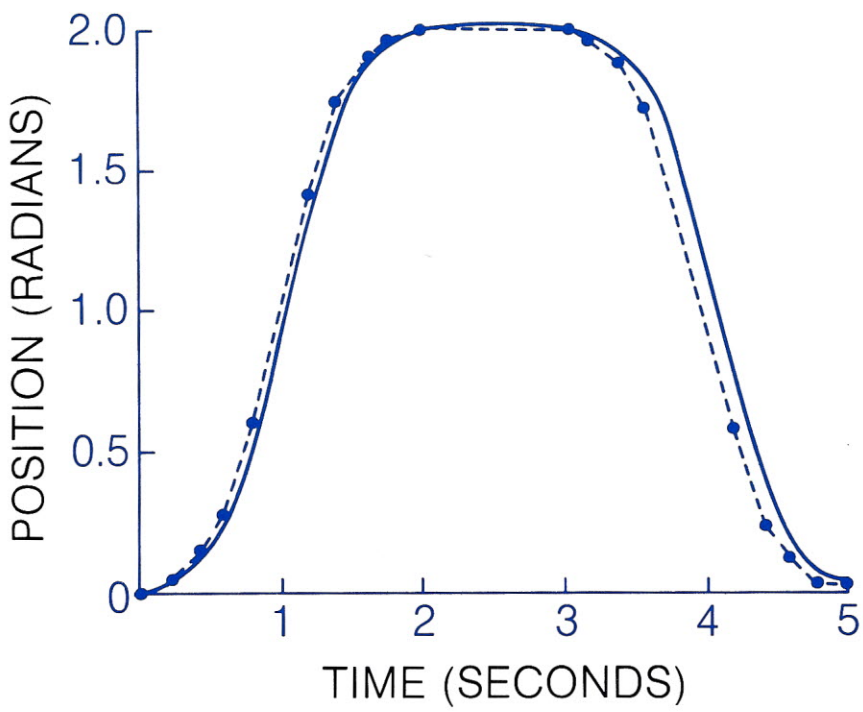


Figure 9. Response of the system to a bell-shaped trajectory.

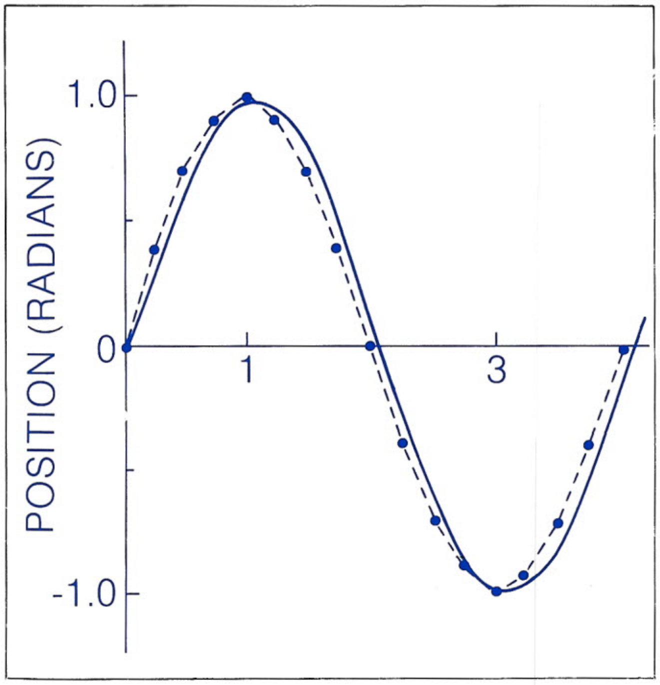


Figure 10. Response of the system to a sinusoidal trajectory.

advance. This is possible in many applications where the program is essentially "canned," but not in others, such as conveyor tracking, where the path is continuously modified.

With the additional computational power achievable today, many robot manufacturers are moving in the direction of performing their servo calculations in Cartesian space, thus providing a capability for controlled interaction with moving objects, and introducing a whole new set of interesting concepts. [3,4] These will be addressed in future articles in ROBOTICS AGE.

References:

[1] Berthold K. P. Horn, "Kinematics, Statics, and Dynamics of Two-Dimensional Manipulators," in *Artificial Intelligence: An MIT Perspective,* Winston and Brown, eds., MIT Press, Cambridge, MA (1979).

[2] R. M. Phelan, 1977. *Automatic Control Systems.* Cornell University Press, Ithaca, New York 14850.

[3] R. Paul, "The Mathematics of Computer Controlled Manipulators," Joint Automatic Control Conference, San Francisco, 1977.

[4] K. Takase and T. Okada, "Robotics Research in Japan," *ROBOTICS AGE*, Vol. 1 No.2 (Winter 1979).

[5] Maroof Mian, "Trajectory Control of a Robot Joint with a Microcomputer," MS Thesis, EE Dept., NCSY (1978).

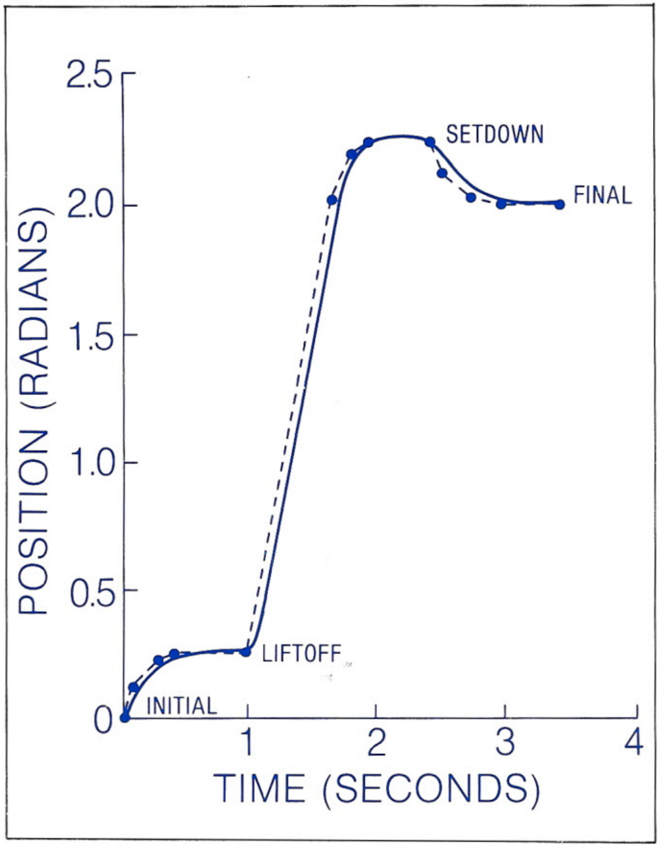


Figure 11. Response of the system to a typical joint motion.

Acknowledgements

The experimental portion of this work was sponsored in part by the National Science Foundation (grant ENG-77-06771). The computer programs were written by Maroof Mian while he was a student at NCSU, and more detail can be obtained from his thesis. [5] The concept of piecewise linear velocity control originally evolved from a suggestion by Richard Paul. Many useful suggestions about the control theory involved were provided by my colleague at NCSU, Bill Gruver.

About the Author

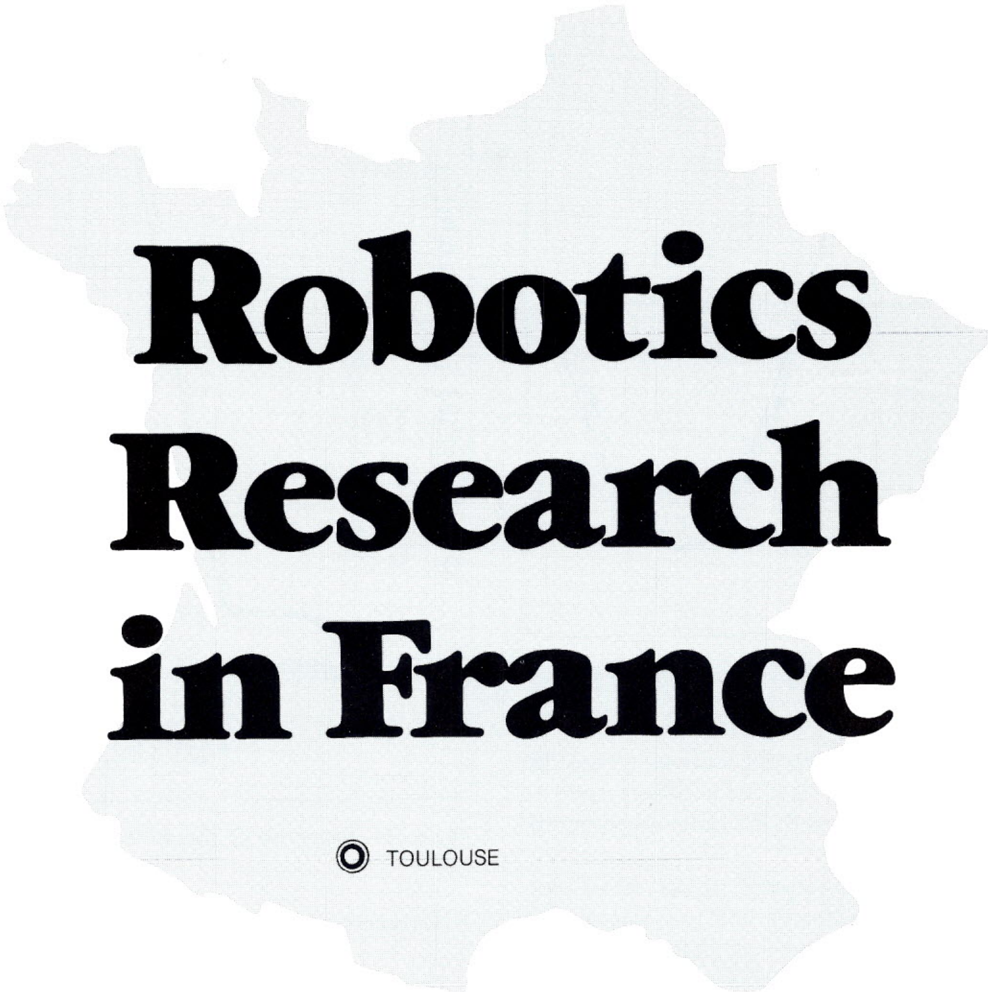
Dr. Snyder is an assistant professor of Electrical Engineering at North Carolina State University, where he is an associate director of the image analysis group.

At present he is on leave with the West German space agency (Deutsche Forschungs-und Versuchsanstalt fuer Luft-und Raumfahrt). His specialties are computer applications to automation and in particular to analysis of images.

Prof. Snyder has taught courses in applications of microprocessors at IBM, Raleigh, and such faraway places as Warsaw, Poland. His current assignment in Germany involves research into use of microcomputers in the real time analysis of television images.

He is a member of the IEEE, the ACM, a senior member of the SME, and a charter member of the Robot Institute of America.

Robotics Research in France

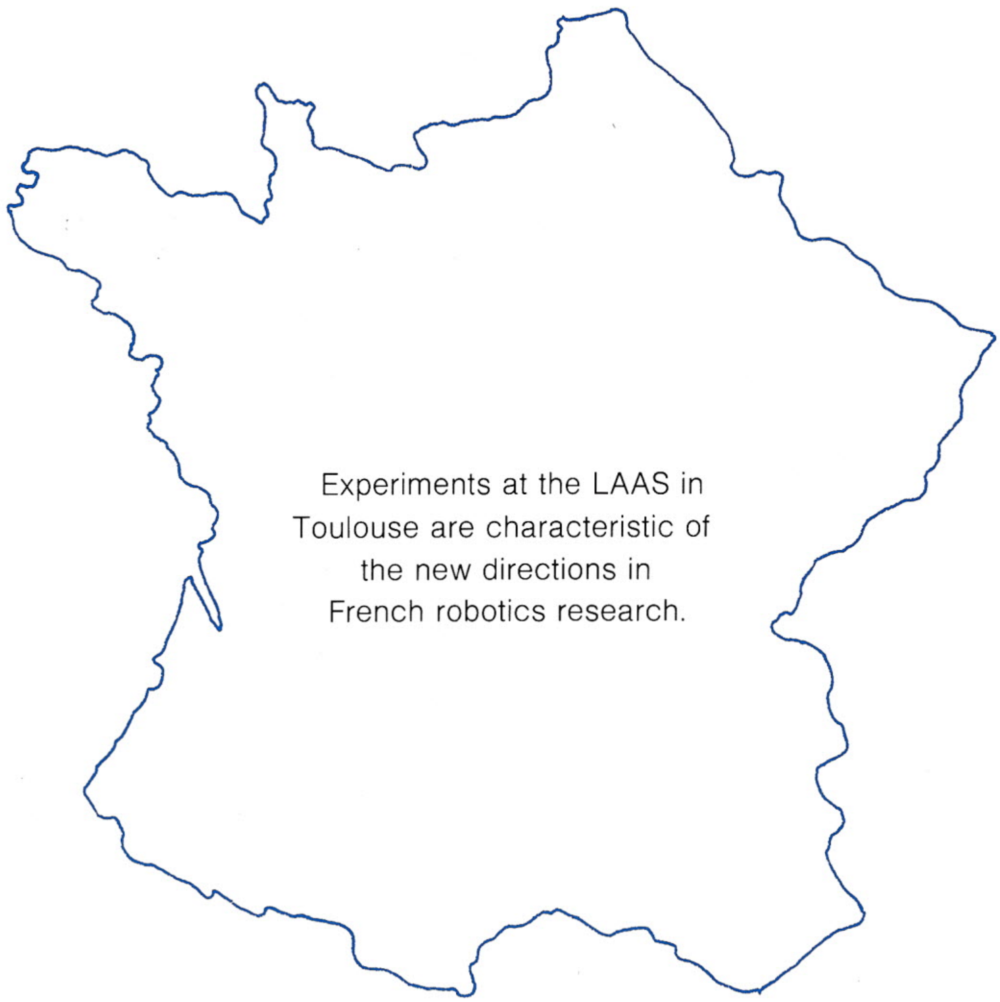


Roland Prajoux
Laboratoire d'Automatique et d'Analyse des Systēms (LAAS)
C.N.R.S., Toulouse, France

Founded in 1967, LAAS has since that date conducted research in the fields of semiconductor components, automatic control, and computer science, as well as in robotics. In robotics, the Laboratory was at first interested in the dynamics of manipulators and in sensors. For example, as early as 1974, an artificial skin was introduced [4]—a sensor whose utilization is now widespread. In 1976, a new effort in robotics research began at LAAS directed exclusively toward the development of "intelligent" robots—I was tempted to write "true" robots.

At this point, I have no intention of being caught in the trap of defining once again what could or should be a robot—I mean an intelligent robot. However, I believe it is necessary to specify what kind of robot systems we are interested in. A good definition I have heard of is: "Systems capable of perceiving and recognizing their environment, having the ability to react to changes or perturbations in this environment, and, by using some knowledge of shapes and forms, capable of doing anthropomorphic movements and manipulations." This definition calls for a few comments: 1) The fact that the robot is acting in the real world must be stressed—2) The word anthropomorphic is not fundamental to the concept: the *goal* of the system or the *consequences* of its actions can have some human-like character, but it is not relevant to consider if the *way* the system is doing them is anthropomorphic or not—3) Some important elements are missing in the definition: intelligent robots can be assigned various tasks, they make decisions by themselves and they communicate with the human operator in a sort of "natural" way.

Given this framework, I can now introduce the work that we are currently carrying out at LAAS. Our program follows two guidelines: 1) Conducting experiments is of prime importance. In short, "there is no robotics without robots!" Indeed, it is not uncommon to meet people claiming that they are involved in robotics, but there are no



robots to be found in their laboratories. 2) The mission of the Laboratory is to do basic research, all the while being concerned with short-term or medium-term industrial applications.

Among the three main functions needed for an intelligent robot are: a) perception, b) decision and planning capabilities using knowledge about the world, c) communication. We are interested in the first two. We deal with perception problems both by studying sensors themselves (artificial skin, ultrasonic and laser sensors, especially) and their utilization. Most of our efforts are directed toward vision, which is, and probably will remain, the most important sensing means in robotics, but also toward ultrasonic and laser perception. The simultaneous and concurrent use of several different sensors, each with its own qualities and deficiencies, is a problem arising in the case of multisensor robots, to which we pay careful attention.

In this paper, instead of trying to cover the whole range of robotics research at LAAS, we have chosen to deal with three experiments which will provide a general picture of our program. The first two experiments (the mobile robot HILARE and a blocks-world assembly robot) are not directly linked to an industrial application; they are a part of long-term basic studies. The third one (a part-mating experiment), conversely, was undertaken with a precise industrial application in mind.

However, before describing these specific projects, some time will be devoted to the important issues of the computer architecture and the planning software organization of intelligent robots. Our ideas about these issues have, of course, strongly influenced the design of the experiments.

Computer Architecture and Planning Software Organization

Sophisticated robot systems with complex tasks to achieve must have powerful and efficient data processing

capabilities. To organize these capabilities, several distinct ideas may be considered:

- Modularity is very desirable. As a consequence, sensors and effectors should have their own microprocessor control. All the problems related to a given sensor or effector are therefore solved once and for all. The combination of sensor/effector and microprocessor becomes a self-contained entity capable of performing the task. This task can be activated by other parts of the system using a command language of a significantly higher level than a direct command of the sensor/effector.
- A dedicated local minicomputer is necessary to control the system, to perform medium-size computations, and to interact with the user of the system.
- Some data processing tasks to be performed are very complex (for example, high-level planning) and thus necessitate the use of a very large computer. The latter can reasonably be accessible only on a time-shared basis.

We have therefore defined a computer architecture comprising 3 levels of processing resources. However, the way this architecture is being used in the two experiments we describe is not very usual: the level which monitors the robot system is not the highest on the computing power scale. The most powerful level is seen only as a resource to which access can be gained occasionally, while the local minicomputer level is constantly in charge of the system. In case of a failure of the large time-shared computer or if the link with it is cut, the local computer can undertake the

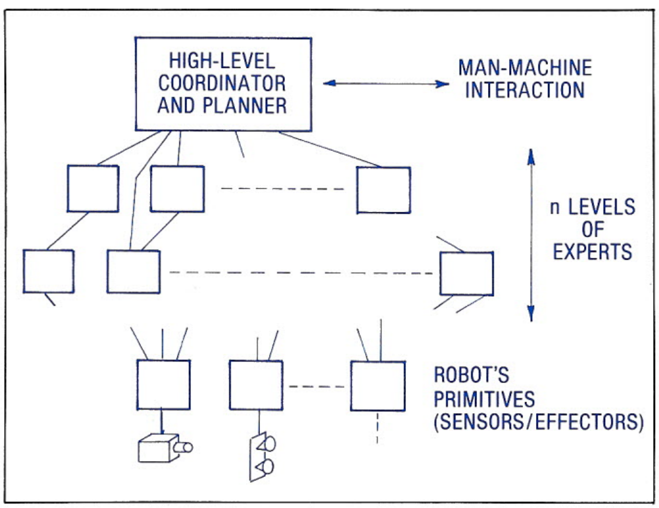


Figure 1. Distributed decision-making structure.

task of controlling the system in some degraded fashion, thus providing a solution to the problem of a gracefully degradable robot.

The architecture defined above does not come from elaborate considerations. I would say that it is an architecture valid almost *de facto*.

Let us now consider the planning software. We have chosen to use a distributed planning system. The reasons for this are:

- A software structure similar to the hardware structure provides better efficiency and robustness. For example, a timely system response can be obtained through the simultaneous treatment of various decisions.
- A centralized, and thus general, decision-making system may be very ineffective to solve some of the problems arising in a robot task. We strongly believe, therefore, in the importance of task-specific (domain-dependent) planning systems. What they lose in generality and flexibility is gained in effectiveness. Indeed, it is possible to completely tailor the planner to the task to perform, including the model of the world, the heuristics and the specific properties of the problem to be solved.
- Besides efficiency, a task-specific planner also has the advantage of the program size being directly dependent on the problem complexity, whereas a general-purpose planner would hardly be small. On the other hand, when it comes to man-machine communication, or to changing something in the robot's task, or to some learning capability, the flexibility of a general-purpose planner is necessary.

The decision-making structure embodying the above ideas is hierarchial and modular. It consists of independent expert modules and a high-level coordinator cooperating in a multi-level structure. The experts are specialized in various domains. Each one has its own data base and world model. Every expert can be used as a primitive, or an operator, by a higher level expert. The coordinator is a general-purpose planner.

The above structure leads to several research problems—among them: how to design the coordinator, perhaps adapting it to the needs of a robot instead of preserving a truly general planner?; is it possible to derive common rules for the design of different experts?; what kind of information should be exchanged between two experts, bearing in mind that they have different world models?

The distributed decision-making structure is summarized in Figure 1.

I. The Mobile Robot HILARE*

The HILARE project started in September 1977 as an attempt to provide the LAAS robotics group, as well as several other teams interested in this field in Toulouse, with a flexible and powerful experimental support for advanced research work in robotics.

The choice of an autonomous mobile robot was made mainly for two reasons: 1) capacity to provide a variety of problems at different levels of generality and difficulty in a large domain—including perception, learning, decision-making, communication, etc., all of which have to be considered within the scope of the specific constraints of robotics: on-line computing, cost considerations, operating ability and reliability; and 2) the possibility of designing a modular system which could be incrementally built and yet providing interesting and useful research opportunities at every step.

We consider HILARE our most important experimental facility, and we intend to use it to test and demonstrate most of our theoretical work for several years to come. We emphasize that HILARE has no practical purpose in itself; yet we are deeply concerned at every stage of our work with methods, software and instrumentation for advanced applied robotics. This was the only way for us to guarantee a sufficient level of generality for the project.

Physically, HILARE is a triangle-shaped cart with three wheels, the front wheel being a free caster. The cart is built in three levels. The lower level contains the locomotion components (stepping-motors and batteries). The intermediate level contains the electronic systems and the upper level supports the main sensors and, in the future, will be equipped with a manipulator (see Figure 2).

The locomotion of the vehicle makes use of stepping-motors for each drive wheel. They are controlled by a microcomputer which, in our structure, constitutes the motion expert. [1]

HILARE is connected to the MITRA 15 minicomputer monitoring its operation via a full-duplex radio transmitter.

At the present time, the perception of HILARE consists of a video camera complemented by a laser range-finder, ultrasonic sensors and a triangulation radar system. [3] The latter system is composed of two infra-red emitter-receivers on a revolving base powered by step motors. The room is equipped with three retro-reflecting "corners" (i.e., beacons). Measurement of angles is made by counting the control impulses. A system of three beacons per "corner" allows parasitic echoes to be eliminated. In addition, one corner was made different from the other two so as to uniquely define the origin.

This system, which is controlled by a microcomputer, allows the robot to determine its location within a few centimeters. Although the triangulation positioning system is only temporarily used on HILARE, such a technique is perfectly valid from an application point of view. One can imagine, for example, placing the reflecting beacons in a warehouse where a mobile robot would be operating.

The vision system, which is still under development, makes use of an image coded with 8 grey levels. The laser range-finder is triggered for a limited number of "shots" to obtain the distance of relevant regions in the image (see Figure 3). The structure of the combined mounting of the camera and the laser is shown in Figure 4.

HILARE is also equipped with ultrasonic sensors arranged as shown in Figure 5. We intend in the near future to increase the number of these sensors, at least at the front of the robot, to improve the capabilities of the

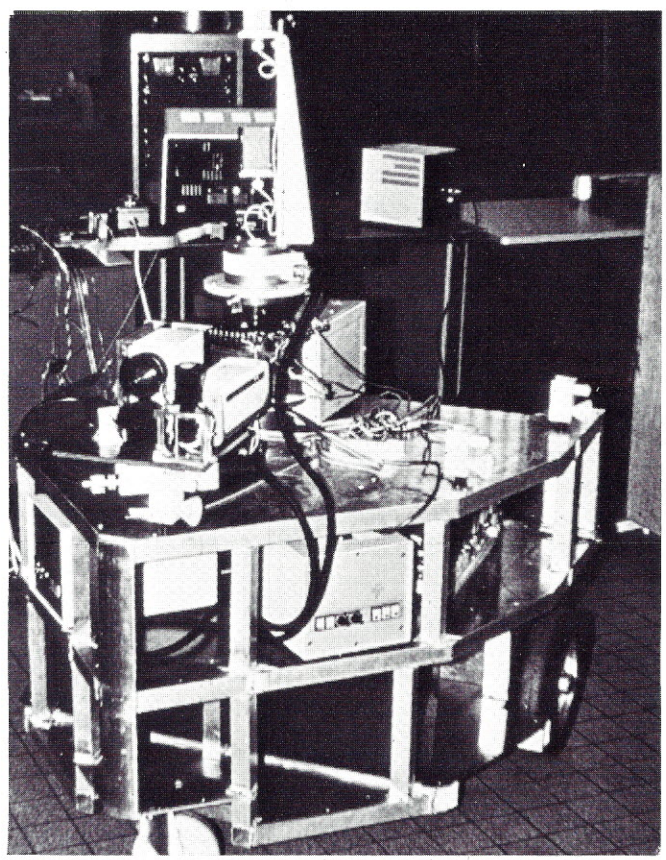


Figure 2. The mobile robot HILARE.

*HILARE: "Heuristiques Intégrées au Logiciel et aux Automatismes dans un Robot Evolutif."

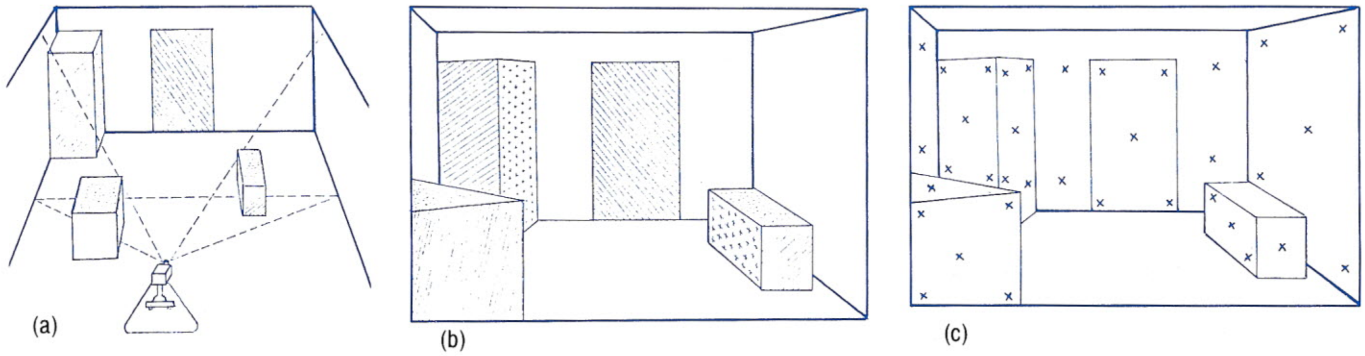


Figure 3. 3-D perception: a) view of the room, b) as seen by the grey-level camera, c) examples of laser "shots."

ultrasonic sensing. The data so obtained will be utilized to provide: 1) a proximity alarm; 2) a general use range-finder; 3) navigation in special cases (along a wall, or to by-pass an obstacle, for example) without using the vision.

A schematic view of HILARE, showing all the afore-mentioned sensors, is given in Figure 6.

We now consider the distributed decision making of HILARE, which follows the general scheme given above. [7] The design can be viewed as decisions through multiple cooperating expert modules together with a high-level coordinator in a hierarchical structure.

The expert modules have their expertise in a variety of overlapping domains (e.g., object identification, navigation, exploration, itinerary planning). The modules consist of 1) specialized and redundant knowledge bases, 2) algorithms and heuristics, 3) local error-processing capabilities, and 4) communication procedures. Modules may have access to one another as primitive action units. The coordinator activates modules based on an analysis of the current situation.

We are designing and programming a relational-level planner at an advisor level. A plan at this level is a flexible dynamic structure which coordinates the achievement of desired goals.

Due to the incremental and open-ended nature of HILARE's design we consider *Production Systems* (PSs) as a viable research tool for relational-level problem-solving. Thus, we are writing the planner terms of a PS architecture in the spirit of M. Rychemer. [11]

The expert module for navigation [7] considers the basic problem of moving a robot from an initial location (R) to the target (G) within a given place. This involves obstacle avoidance, pathfinding, and search trajectory minimization.

Currently, obstacles are defined as polyhedral objects, whose floor projections fully determine the navigation problem, and they can be located either by initial information of by robot perception (see Figure 7). Each obstacle projection is represented as an ordered list of segments in a counter-clockwise sequence.

Empty areas are defined as convex polygonal cells which may include obstacle segments. A trajectory within such cells can be considered as a straight line between entry and exit segments. Two adjacent cells have connectivity through common segments which are traversable by the robot. Thus, a connectivity graph provides the structure necessary for pathfinding.

This graph is fairly similar to the connectivity graph describing the topology of a group of rooms, which will be used by the expert module for intinerary planning. We expect in both cases to be able to implement some sort of learning process, which is useful when the robot is discovering an unknown environment by itself.

To conclude this presentation of HILARE, it is noted that the ongoing work on this robot includes:

- sensory integration,
- implementation of a multi-microprocessor structure,
- developing various expert modules,
- vision algorithms with grey levels,
- design of the high-level planner using pattern-

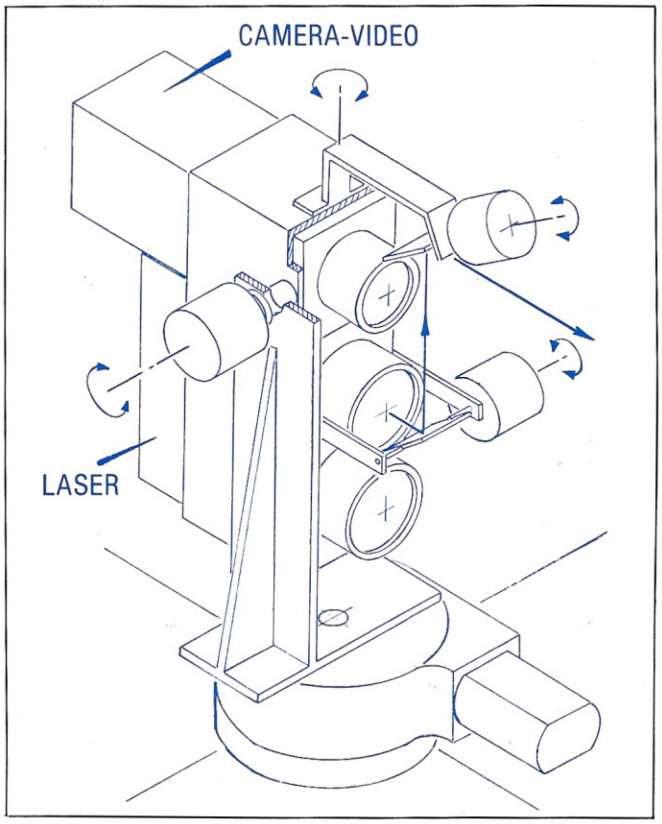


Figure 4. Laser range-finder and camera mounting.

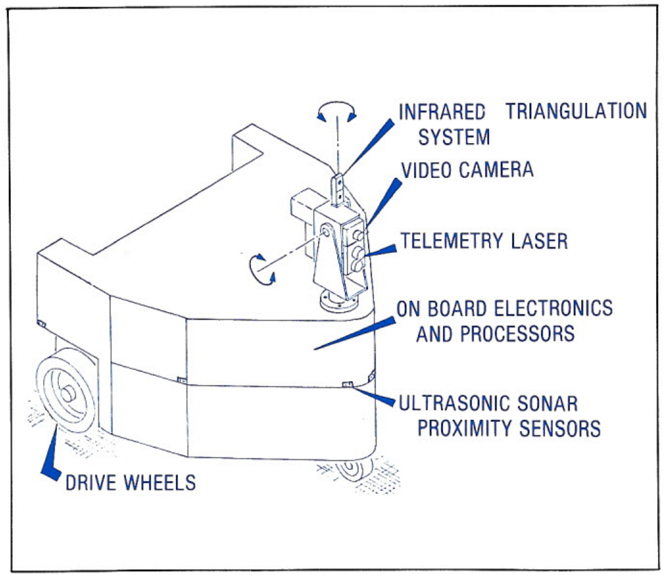


Figure 6. Schematic view of HILARE's current hardware arrangement.

directed inference systems (PDIS) techniques.

- implementation of a control by voice, with the collaboration of two other laboratories (L.I.M.S.I.-Paris for the voice decoding and L.S.I.-Toulouse for sentence understanding).

II. Blocks-World Assembly Experiment Using Task-Specific Planning.

Flexible Automated Assembly Systems are among the most challenging problems that many groups working in Robotics currently encounter. Several studies designed to contribute to this field of research are being carried out at LAAS. This section will describe one study which is especially concerned with two important aspects of robot assembly problems: System Integration and Plan Generation. [10]

Blocks-world models have often been noted as relevant and interesting research vehicles by different authors. [12-13] In our experiment the assembly robot system operates in a discrete parallelepiped world containing any finite number of identically sized blocks (7x7x3.5cm). Cubes (as they are called) belong to one of five possible classes; the class of a cube is represented by a pattern on its upper face (i.e., circle, rectangle, triangle, five-pointed star, or eight-pointed star).

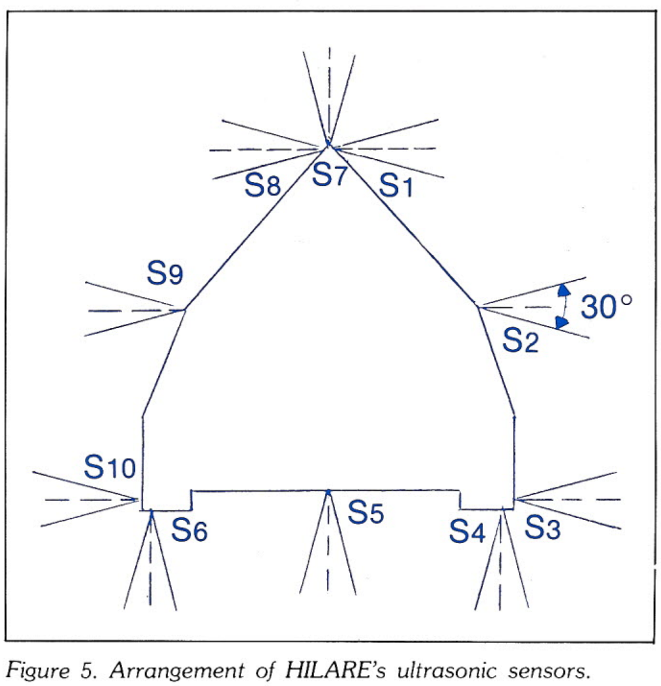


Figure 5. Arrangement of HILARE's ultrasonic sensors.

System Architecture

The blocks-world assembly experiment makes use of a robot called the "X-Y-Z Table" or "transfer robot." Its physical structure (see Figure 8) is that of a rectangular work area above which a manipulator operates. The manipulator has four degrees of freedom X-Y-Z-θ. The first three are translations along orthogonal axes and the last one a rotation around the vertical axis. Various devices can be mounted on the manipulator, ranging from simple magnetic or vacuum grips to sophisticated wrists. In the past, a sensor making use of the LAAS artificial skin was also mounted to perform parts recognition by touch sensing. [2] However, the sensor commonly used is a video camera whose axis is parallel to the vertical axis and which may be moved along the two other orthogonal axes.

The X-Y-Z table is microprocessor-controlled, and the video camera system consists of a Sanyo VCM 2000 video camera (8 MHz bandwidth) coupled to an Aerazur INF 625 image digitizer. The digitizer transfers 6-k bytes by Direct Memory Access to the computer (a MITRA 15) executing the vision software (24,576 points per image at 2 bits per point).

The computing structure consists of 3 levels of processors: 1) an IBM 370/168 remote time-sharing system, 2) a Sems 16-bit MITRA 15 minicomputer (64-k bytes) and 3) an Intel 8080-A microcomputer. The MITRA 15 computer is the system coordinator (see Figure 9); it controls executive functions and the communication between the user, the IBM 370, and the assembly robot. There is a control terminal for the system user and a line-printer for recording diagnostic and execution traces. The software system, written in FORTRAN and assembler, requires 72-k bytes and executes in overlayable segments. It also includes the path-finding and vision software. The remote IBM 370 runs under TSO and, therefore, it considers the MITRA 15 as an ordinary terminal. They are connected by a 300 baud half-duplex line. The IBM 370 executes an

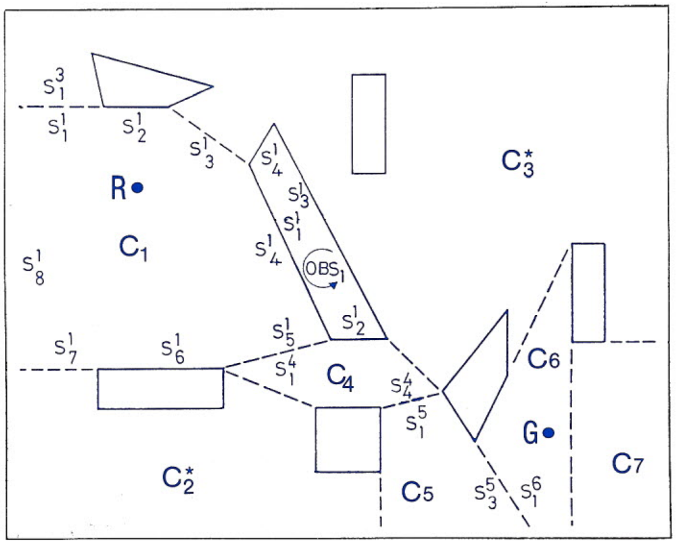


Figure 7. Decomposition of a room into cells by the navigation expert module. R is the location of HILARE, G is its goal. Star-cells (C_2, C_3*) are not completely known.*

interactive APL program (95-k bytes) which generates plans of action for the assembly robot. These two computers form the planning and execution-control function.

The Intel 8080-A micro-processor translates the distance into pulses which control the manipulator's stepping motors. It is connected to the MITRA 15 by a 1200 baud serial line. Its memory consists of 1-k bytes EPROM for program (630 bytes) and 256 bytes of static RAM (20 bytes data plus pushdown stack). The program performs relative displacement with acceleration and deceleration of the stepping motors in X, Y, or Z plus rotation (θ) about the Z-axis and activation of the electromagnet end-effector. The accuracy is 0.18mm per motor step in X, and Y, 0.078mm in Z, and 0.1 degree in θ.

Communication, Vision and Path-Finding Programs

The communication and coordination software resides in the MITRA 15 local computer. The interfaces with the user, the assembly robot, and the remote planning system are completely asynchronous and interactive. Each communication path is characterized by a command language. The user can issue directives which control the MITRA 15 utilizing a simple command language.

Communication with the planning system is controlled by a special module which interacts with the planner. This module is designed to react to the character string sent by the APL system when this system is ready to accept an input. Communication language protocols are defined between the two systems, which include error-recovery procedures that are either automatic or user controlled.

The local MITRA 15 computer also runs the vision software. This is a temporary feature of the system. As soon as possible, we intend to have a modular vision subsystem, which will employ hardware contour extraction.

Perception requirements in the scope of our system are to locate and identify the class of blocks in order to update the state of the world model used by the planner.

The five identifying patterns on the upper faces of the blocks were designed to permit a reasonably simple classification. Digital image processing provides barycenter location and contour extraction. Contour features are then used for pattern recognition. The vertical location of a block is determined by analyzing image shifts produced by programmed changes in camera positioning. In this way, the vision task for the blocks-world experiment acquires a stereo vision capability.

The path-finding system realizes the functions of path calculation and manipulator movement in the universe of the assembly robot. The system (RTMOVE) depends upon the following hypotheses: 1) a discrete blocks-world model, 2) that a cube can be transported between two

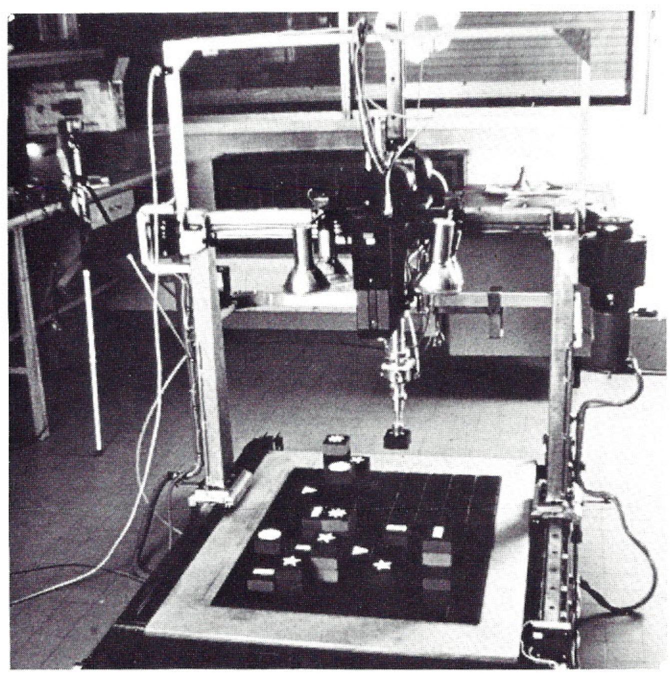


Figure 8. General view of the X-Y-Z table.

stacks of cubes, and 3) that stacks of cubes have a maximum height, so that the manipulator can traverse any construction while carrying a cube. RTMOVE executes the following algorithm:

1) Validate path specification,
2) Plan and execute path trajectory to initial location,
3) Grasp cube,
4) Plan and execute path trajectory to destination, and
5) Place cube.

For each step of the algorithm there are interactive error-recovery procedures.

The path trajectory planning is performed by a version of Lee's algorithm. [9-10] The system produces a path in 3 dimensions that minimizes the "city-block" distance: (| X | + | Y | + | Z |).

A final procedure transforms the vertical path segments, combining them to maintain minimal city-block distance and efficiently reduce changes of direction.

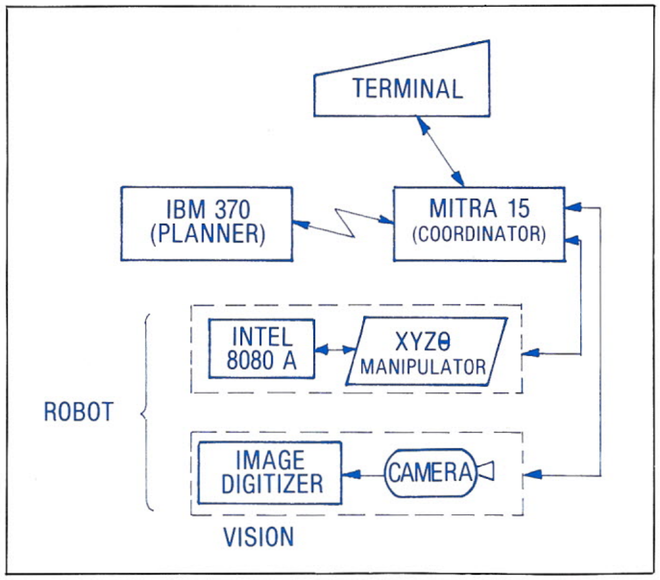


Figure 9. Architecture of the blocks-world assembly experiment.

Task-Specific Planning

The choice of an assembly task using rectangular blocks may at first seem to be questionable, because of the simplifications introduced compared to true industrial assembly. We deliberately made this choice for the following reasons:

- In contrast to most blocks-world studies, we did not make a simulation but a real implementation and thus the general problems involved in system integration have been retained;
- blocks-world problems lead to non-trivial "reasoning," which means that they are relevant for demonstrating the efficiency of task-specific planning.

The problem is to accomplish a construction, specified to the robot, by transforming an initial universe containing an unknown construction.

Data to be supplied to the planner ZPLAN consists of the following items:

- Work-area size. The work-area size characterizes 3 levels of difficulty as follows. If it is large with respect to the initial and final constructions, the problem is an *easy* one. A solution may always be found by 1) simply spreading out the blocks outside the specified final construction limits, 2) discovering their classes

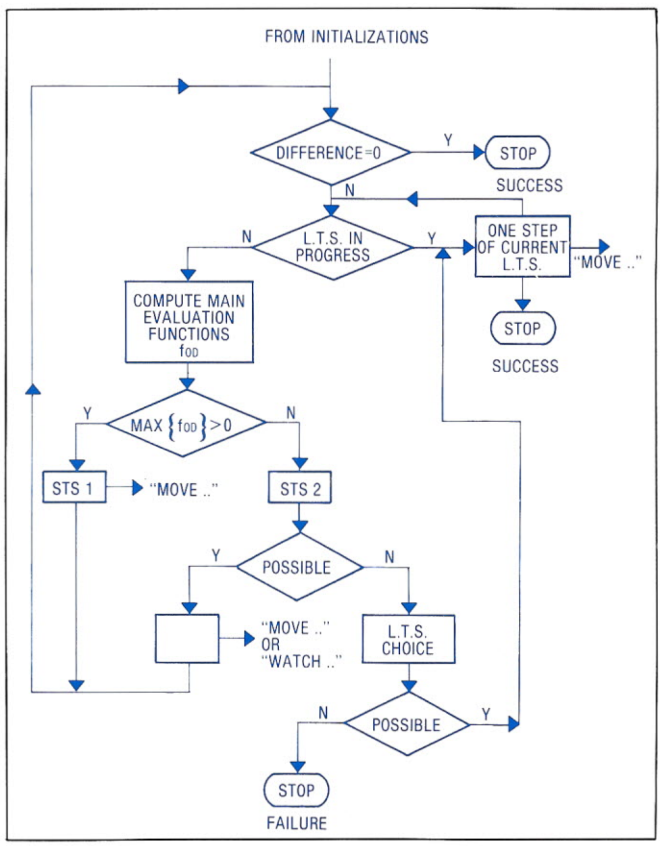


Figure 10. Plan generator flow chart.

using vision, and 3) assembling the goal construction.

- If the size allows enough free space for intermediate constructions, but the plan is not simple to achieve, the problem has *medium* complexity.
- If the initial and final constructions are greatly overlapping, and few temporary free locations are available, the goal is *difficult* to accomplish.
- Available number of cubes for each class. This information allows the planner to test if a specified goal is possible to achieve before starting. The planner can also in certain cases deduce the identity of unknown cubes, thereby economizing on visual processing, as well as manipulating unknown cubes based on the probability that they are members of a given class.
- Goal description. The final construction is specified (in a Cartesian coordinate system) by the desired contents of certain locations. The contents can specify:
  1. a cube of a given class,
  2. an unspecified cube,
  3. empty, or
  4. unspecified (empty or any cube).

For reasons of clarity, the planner isolates the final construction from surplus cubes by emptying locations all around it.

The planner has two *operators* to act in the real world:

$$\text{MOVE}\quad X_1\,Y_1\,Z_1\quad X_2\,Y_2\,Z_2$$

which specifies the origin and the destination of a cube to be moved, and

$$\text{WATCH}\quad X_1\,Y_1\quad X_2\,Y_2$$

which limits the region to be watched (i.e., rectangle defined by the diagonal X_1, Y_1, and X_2, Y_2).

ZPLAN consists of procedures using data redundantly embedded in arrays. Speed is thereby gained at the price of increased memory, as is usual in such a case. Although specific, ZPLAN is designed using general basic principles, namely the concepts of *difference* and *evaluation functions*. [5-6]

- A *difference* is some measure of the "distance" between the current and the desired states of the world. The planner makes use of *several* differences.
- Consistent with the differences, several *evaluation functions* are defined. The main one for each valid operator MOVE O,D, gives the variation of distance due to the application of MOVE O,D, i.e., the consequences of removing the cube in O and filling the location in D.
- A *decision procedure* deals with the case of unknown cubes and/or unspecified locations to decide if the goal is achieved for them. It ensures that an intentionally built-up subset of the goal, containing such cubes and locations, can never be modified later.
- ZPLAN utilizes either short-term (STS) or long-term (LTS) strategies to achieve the goal.
- If the maximum of the main evaluation functions is *positive*, STS1 is selected. A short-term strategy is adequate here because it will bring the system nearer to the goal (i.e., D_1 diminishes in one move).
- If the maximum is *negative*, the system makes a choice between STS2 and two long-term strategies LTS1 and LTS2.

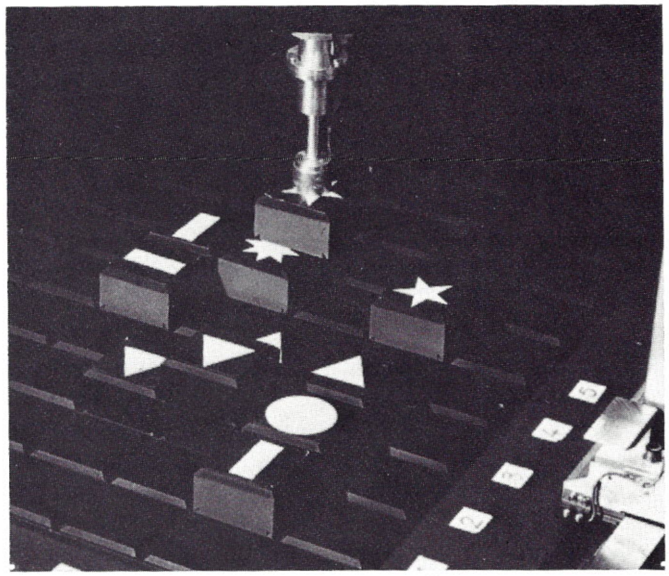


Figure 11. Example of blocks manipulation.

The flow chart of ZPLAN is depicted in Figure 10.

Conclusion

The overall performances of the blocks-world assembly experiment are excellent, confirming the validity of the concepts described at the beginning of this paper.

The planner is capable of solving complex problems efficiently. A rather lengthy plan involving a 90-location

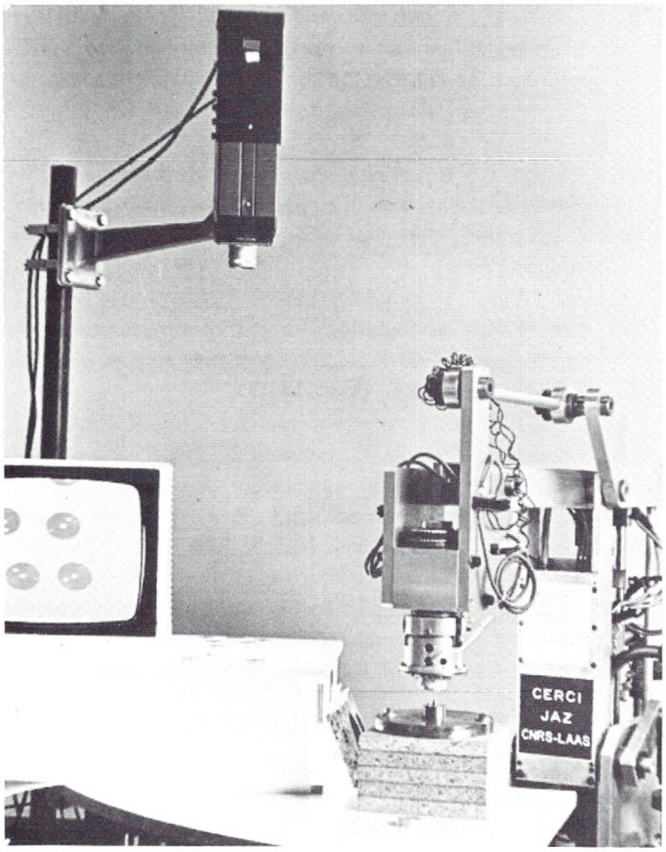


Figure 12. View of the part-mating experiment.

world and 26 cubes, achieved with two WATCH operators and 25 MOVE operators, takes only 10 CPU seconds on the IBM 370-168.

Figure 11 shows the actual manipulation of a block.

These experiments demonstrate an "intelligent robot" doing tasks in the real world by itself, achieving the prescribed goal by using its sensors and effectors and its "reasoning programs."

III. A Part-Mating Experiment Using a Robot with Vision and Force Control

This work was done with the collaboration of two industrial companies, JAZ, which designed the robot arm, and CERCI, which produced the software. The Laboratory was responsible for the overall conception, the sensor integration and the design of a special compliant wrist. [8]

The purpose of this research was a feasibility study for a robot capable of loading and unloading a machine-tool, the parts being presented by a conveyor belt. The study took place in the context of small-size mechanical parts (see Figure 12).

The most important points that we took into consideration were:

- Recognition and acquisition of parts. We used a "training-by-showing" approach, employing a vision system similar to the Vision Module of S.R.I. International*, although significant differences can be found between the two systems. We use a sequential classification procedure, performing pattern recognition for N objects based on M features, structured in a deterministic decision tree.
- Part-mating. After acquisition, the part has to be inserted in a jig located on the machine-tool. This operation is, in fact, a part-mating task. The accuracy

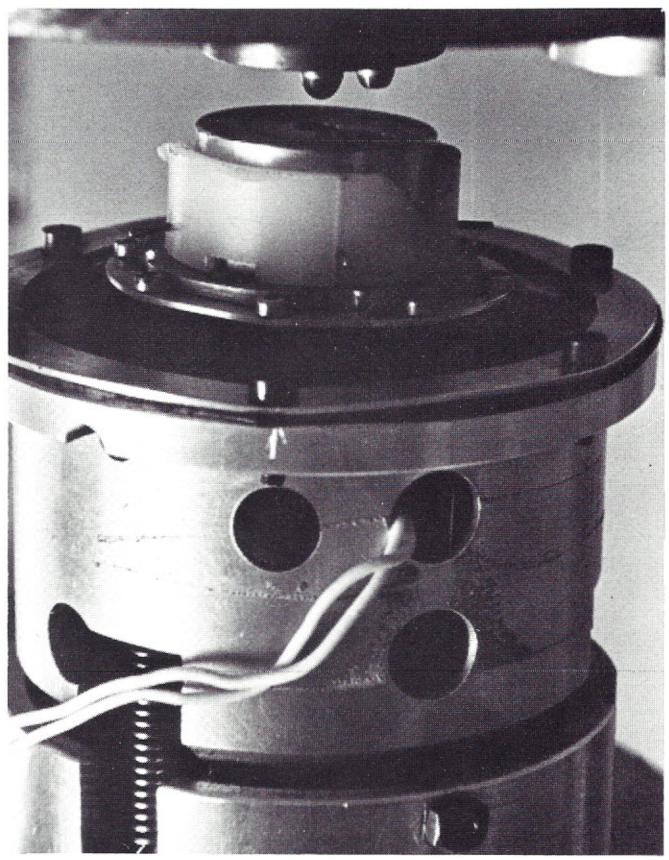


Figure 13. The compliant wrist.

**Editor's Note: See "Industrial Robots—1979"* ROBOTICS AGE, *Vol. 1, No. 1.*

of the approach phase is about 0.3 millimeter. To achieve the part-mating operation properly, we have designed a compliant wrist. The wrist is rigid along the insertion direction and compliant along the others in a passive way (see Figure 13).

The experiment has been carried out very successfully. The current work is concerned only on minor improvements to reduce the operation cycle time.

IV. Conclusion

In this paper, I have described three experiments conducted at LAAS in the field of Robotics Research. Although they do not cover the whole work that we are currently pursuing, they give a good illustration of the way we are doing our research in robotics: we try to maintain a balance between long-term basic research and medium-term research oriented toward industrial applications.

We all find the field of robotics a very exciting one, and we have the feeling that we are participating in something really important for the future of our society.

References

[1] Bauzil, G., and Ribes, P., "Robot HILARE: réalisation d'un robot mobile," Technical report LAAS—MCI 79.I.39, Sept. 1979.

[2] Briot, M., "La stéréognosie en robotiqe, application au tri de solides," Doctorat d'Etat thesis, Université Paul Sabatier, Toulouse, 1977.

[3] Briot, M., Talou, JC., and Bauzil, G., "Design of a multisensor system for the autonomous mobile robot HILARE," (To be published in *Industrial Robot),* 1980.

[4] Clot, J., Lagasse, J., Falipou, J., "Capteur d'empreintes (peau artificielle)," Patent A.N.V.A.R. N° 74.18.944, May 31st, 1974.

[5] Ernst, G.W., and Newell, A., *GPS: A Case Study In Generality and Problem Solving,* Academic Press, NY, 1969.

[6] Fikes, R.E. and Nilsson, N.J., "STRIPS: a new approach to the application of theorem proving to problem solving," *Artificial Intelligence,* 2.1971.

[7] Giralt, G., Sobek, R., and Chatila, R. "A multi-level planning and navigation system for a mobile robot; a first approach to HILARE," Proc. 6th Int. Joint Conference on Artificial Intelligence, Tokyo 1979, pp. 335-337.

[8] Giraud, A., "Robot de chargement et déchargement d'un poste de travail sur une machine à plateau tournant." Final report, contract DGRST 78.7.0402, (to be published) 1980.

[9] Lee, C.Y., "An algorithm for path connections and its applications," IRE Trans. on Electronic Computers, vol. EC-10, pp. 346-365, Sept. 1961.

[10] Prajoux, R., Sobek, R., Laporte, A., and Chatila, R., "A robot system utilizing task-specific planning in a blocks-world assembly experiment," Proc. 10th Int. Symp. Industrial Robots, Milan, March, 1980.

[11] Rychener, M., " Production systems as a programming language for artificial intelligence applications," Ph.D. Thesis, Carnegie-Mellon University, Dec. 1976.

[12] Winograd, T., *Understanding Natural Language,* Academic Press, 1972.

[13] Winston, P., *Artificial Intelligence,* Addison-Wesley, 1977.

Glossary

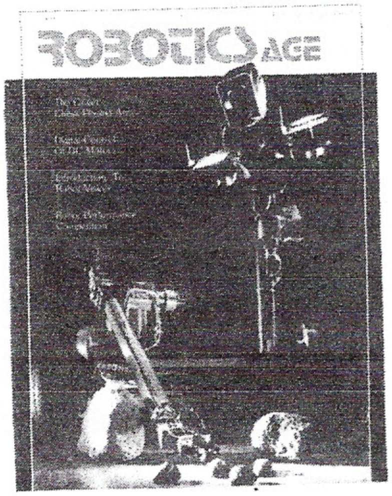
heuristic: *a technique used in a problem-solving system that may improve the program's performance. The term often refers to the methods used to organize the search for the solution to the problem.*

production system or pattern-directed inference system: *a method of programming in which a collection of "if-then" rules define the system's behavior. Whenever the "if" part of any rule is satisfied by facts in the database, its consequence is automatically performed.*

world-model: *a database used by a problem-solving system in which the facts represent the program's beliefs about the state of its environment.*

Robotics Age wishes to acknowlege the kind assistance of Dr. Georges Giralt, director of the LAAS robotics program, without whose help this report would not have been possible.

Sold Out!



Vol. 1, No. 1, Summer 1979

Due to the overwhelming demand for ROBOTICS AGE, the first printing of our Summer 1979 issue has quickly sold out. However, we are able to offer a limited number of copies for distribution. These are available in two categories:

- The *Collector's Edition*—The first 200 copies from the first press run have been individually numbered and signed by both the Editorial Director and the Publisher. These registered copies are available for \$20.00, plus \$1.50 for first-class postage and handling. (First issues of other popular magazines have often become worth hundreds of dollars.)

- Imperfect Issues—There are also a small number of issues which we held back due to various flaws such as loose pages, ink streaks, etc. These have been checked to ensure that the entire contents are complete and legible and may be ordered for \$2.50 plus \$1.25 for shipping and handling.

Send your orders to:

Back Issues
ROBOTICS AGE Magazine
P. O. Box 801,
La Canada, CA 91011

COMING SOON

IN ROBOTICS AGE

- Low cost video interface
- Introduction to speech recognition
- Applications of the Cuk Ćonverter
- Introduction to robot reasoning
- Methods of multiple CPU architecture

and MORE!!!

We regret that the process of converting to a new system has delayed the processing of our reader service cards. We apologize for any inconvenience this may have caused.

PRODUCT NO.	PRODUCT
1	Cincinatti Milacron, T^3, front cover
2	Digital Video, CAT 100, p.1
3	Atchley Controls, Servo valves, p.1
4	Industrial Robots, Newsletter, p.2
5	Terrapin, Robot Turtles, p.27
6	Robotics Industry Directory, p.27
7	Spatial Data, EyeCom, p.39
8	Microbot, Mini-Mover 5, p.44
9	Data Translation, Analog Interface, p.44
10	Robotics Technology, consultants, p.45
11	Hughes, Omneye, p.45
12	Conn. MicroComputer, BSR Interface, p.46
13	Cincinnati Milacron, T^3, p.46
14	Gallaher Research, Grivet, p.47
15	Astek, Accomodator, p.47
16	Technical Insights, Newsletter, p.48
17	Datel Intersil, A/D Converter, p.48
18	Bits, "Artificial Intelligence", p.51
19	ComputerWorld, Kits & Components, p.53
20	Cybernetic, Stepper Motor Controller, back cover

ARTICLE NO.	ARTICLE
1	Micro-Computer Based Path Control
2	Robotics Research in France
3	Multiple Sensors for a Low-Cost Robot
4	The Robots of AutoFact II

ROBOTICS AGE

P. O. Box 801, La Canada, CA 91011

subscribe today!

NAME

TITLE COMPANY

ADDRESS

CITY STATE ZIP/COUNTRY

Payment enclosed
☐ 1 yr ($ 8.50)
☐ 2 yr ($16.00)

Bill me/Bill my company
☐ 1 yr ($ 8.50)
☐ 2 yr ($16.00)

Charge my ☐ **Mastercharge** ☐ **Visa**

Acct. # ______ **Expires** ______

Foreign
☐ 1 yr ($15 US)
☐ 2 yr ($28 US)

Canada
☐ 1 yr ($10 US)
☐ 2 yr ($19 US)

Please include payment with your order.

READER SERVICE

Keep up to date! ROBOTICS AGE advertisers offer you free of charge and obligation their latest information on their new products...new services...new opportunities. Use the cards on this page. Circle the number shown in the advertisement. Detach and mail. Requests cannot be honored unless zip code is given. ROBOTICS AGE advertisers will send you their latest information. free! This card is valid until 6/30/80.

1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20

NAME

TITLE/COMPANY

ADDRESS

CITY

STATE ZIP COUNTRY

To expedite response, please answer the following questionnaire. Please check only one box per question.

Interest in Robotics?
☐ Home Experimenter
☐ Industrial Applications
☐ Research
☐ Other ______

Type of Business?
☐ Research and Development
☐ Manufacturing (heavy)
☐ Manufacturing (light)
☐ Consulting/Engineering

READERS ADDRESS THE EDITOR

This is your channel to the Editor. More articles will be included in future issues by authors and on subjects which receive high ratings.

ARTICLE NO.	POOR		GOOD		EXCELLENT	
1	0	1	2	3	4	5
2	0	1	2	3	4	5
3	0	1	2	3	4	5
4	0	1	2	3	4	5

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 17539 HOUSTON, TEXAS

POSTAGE WILL BE PAID BY ADDRESSEE

ROBOTICS AGE
P. O. Box 801
La Canada, CA 91011

EXPLORE THE ROBOTICS AGE

The long awaited age of robotics is here.

Already robots with a degree of "Artificial Intelligence" are being installed in factories around the world. A major new industry has been born and its sales are doubling yearly. Universities, companies and governments around the world are funding major new research efforts. Interest in experimentation is increasing and amateur clubs are being set up. A flood of new products are being marketed.

ROBOTICS AGE
P. O. Box 801
La Canada, CA 91011

Subscribe Today

And let ROBOTICS AGE keep you informed of the growth in ALL areas of the THIRD INDUSTRIAL REVOLUTION — ROBOTICS!

If the subscription card has been removed, just send

$8.50 for 1 year

$16.00 for 2 years

To
ROBOTICS AGE
P. O. Box 801
La Canada, CA 91011

"Many people make the assumption that robot experimentation is beyond the resources of the average hobbyist. The simple interfaces described here provide the means, both in hardware and software, for interesting experimentation at minimum cost."

Don McAllister
4709 Rockbluff Dr.
Rolling Hills Estate, CA 90274

MULTIPLE SENSORS FOR A LOW-COST ROBOT

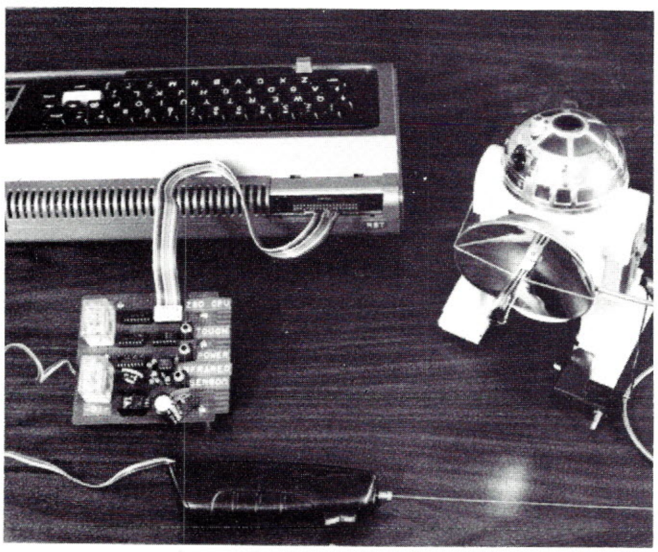


Figure 1. Test setup for a multiple-sensor robot. Both the ADC and the drive output hardware are contained on a single circuit card. (Radio Shack #276-155)

Robot behavior involving multiple sensory functions is a fascinating field of research, for it is in the complex interactions between sensor, control functions, and the environment that behavior similar to that of animal life forms can be produced. Until recently such investigations were well beyond the financial resources of most hobbyists, but with the advent of inexpensive home microcomputers, this is no longer the case.

This article will describe a method of constructing an extremely inexpensive multi-channel analog to digital converter (ADC) that can be used with the Radio Shack TRS-80 computer, allowing programs written in Level II BASIC to acquire input from several sensors. Behavior modes using the sensors are demonstrated through direct computer operation of a radio-controlled toy, in this case a Kenner "R2D2." Although the techniques presented here are for the TRS-80/Kenner-R2D2 combination, they are sufficiently general to be adapted to any microcomputer or test vehicle.

The components of the experimental setup are shown in Figure 1. The interface board plugs directly into the

expansion port of the TRS-80. Output lines from the interface connect to the drive switches in the Kenner R/C transmitter. However, since there is no return radio link, signals from the sensors mounted on the robot must be carried by wires. The use of lightweight coax minimizes the introduction of electrical noise. The sensors shown include a phototransistor infrared detector (used with a spherical collector mirror) and a microswitch touch sensor. Other sensors may also be used, such as a microphone or temperature or pressure transducers.

Interfacing Multiple Sensors to the TRS-80

A key to the simplicity of the ADC interface is that part of the conversion process is performed by the computer software. The interface hardware converts the signal from an analog level to a pulse whose duration depends upon the level. A program loop in the computer measures the length of the pulse by counting the number of loop cycles that the interface signal is high, exiting with the measured value when the signal returns to low. The interface circuit, shown in Figure 2, may be expanded up to six sensors. By using only one 74367 bus driver chip, data from multiple sensors can be input to the computer using only one 555 timer chip per sensory channel.

Referring to the figure, let's consider the functioning of a typical sensory channel. Each 555 timer is configured as a monostable pulse generator. [1] When triggered on pin 2 by a negative-going OUT′ pulse from the CPU, a positive output pulse is generated by on pin 3. The duration of the pulse is determined by the sensor input voltage applied to threshold/discharge terminals of the 555 (pins 6 & 7). The pulse duration should range from 10 microseconds to 30 milliseconds as a function of the input level from the sensor.

This configuration, although easy to build, does impose some constraints on the sensor output. Before the timer is triggered, pin 7 is held near ground by a normally on transistor inside the IC. When the device is triggered, the discharge transistor is turned off, and current from the sensor is allowed to charge the sensor capacitor. The timer output will remain high until the voltage on pins 6-7 matches the reference voltage on pin 5 (2/3 of the supply voltage, in this case 3.33V). Thus, the pulse duration will depend upon both the output level and the output impedance of the sensor used. A low sensor output causes a long output pulse and vice versa. A resistor network or a one-transistor amplifier may be needed to provide suitable output characteristics for a particular sensor.

To allow the CPU to measure the pulse duration, the output of each timer is connected to one bit of the 74367 tri-state bus driver. When the driver receives a negative-going IN′ pulse from the TRS-80, the values of the timer outputs (either logic 1 or 0) are strobed onto the corresponding bits of the computer's data bus. When IN′ is high (IN=0) the driver chip presents an open circuit to the bus to allow other computations to be performed regardless of the timer output values.

In this circuit no I/O address or port decoding is used. Whenever IN′ goes low, the data on the inputs of the 74367 are placed on the CPU bus. Obviously, no other devices may use the same data bits on the expansion port for input unless suitable device selection logic is added. In this interface, up to six 555 timers may be used, each with its own unique timing pulse. The timers are assigned to data bit D_7 through D_2, and their outputs may be read by the computer at any time using the Z-80's "IN" instruction.

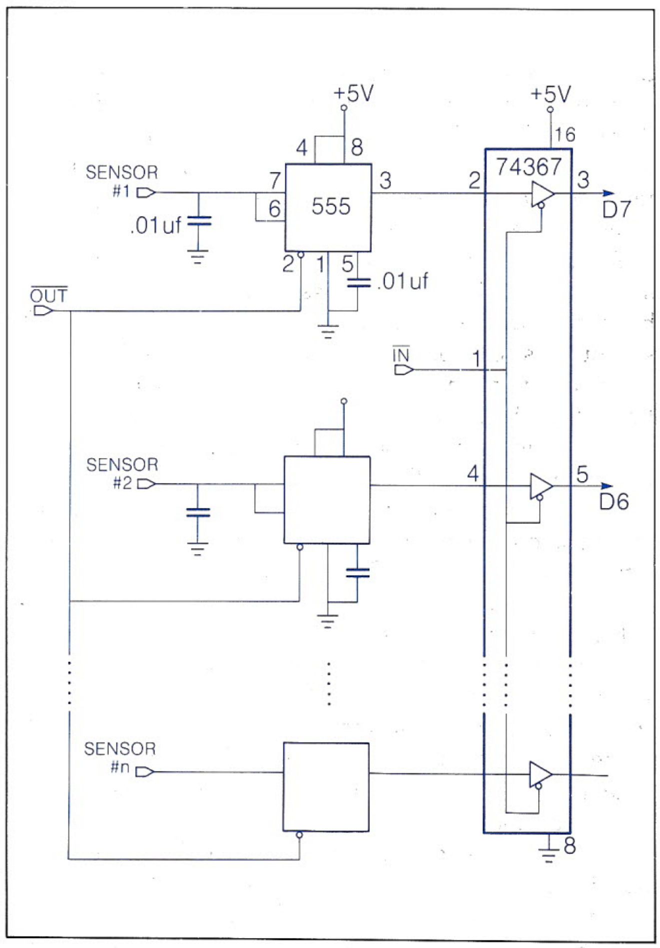


Figure 2. The Analog to Digital Converter (ADC) interface, which plugs directly into the TRS-80 expansion port.

TABLE 1

Multiple Sensor Analog/Digital Conversion Routines in Z80 Machine Language

```
D3 10        A$:     OUT (10H), A      ; trigger A to D converter
CD 7F 0A             CALL 0A7FH        ; initialize converter
44                   LD B,H            ; with most significant byte
4D                   LD C,L            ; least significant byte
03           LOOP1:  INC BC            ; increment counter
CB 58                BIT 3,B           ; limit max converter value
C0                   RET NZ            ; return if value exceeded
DB 10                IN A, (10H)       ; input sensor data D7
CB 7F                BIT 7,A           ; check if D7=1 [see note below*]
ED 43 01 70          LD (7001H), BC    ; store count in 7001H [see note 2*]
20 F2                JR NZ, LOOP1      ; keep counting if D7=1
C9                   RET               ; stop count when D7=0
                                       ; and return to basic
```

*Note: Op-codes for BIT n,A with n=0, 1, 2.., 7 are:

n=7: CB 7F	n=4: CB 67	n=1: CB 4F
n=6: CB 77	n=3: CB 5F	n=0: CB 47
n=5: CB 6F	n=2: CB 57	

*Note 2: To deposit the results at different addresses, change the last two bytes of the LD instruction to the desired address, low-order byte first, so that "ED 43 03 70" will store the result at 7003-4, etc.

TABLE 2

Basic Program for Analog/Digital Conversion and Display

```
10   A$=" SENSOR #1 A/D CONVERSION"
20   B$=" SENSOR #2 A/D CONVERSION"
30   C$=" SENSOR #3 A/D CONVERSION"
40   X=VARPTR(A$)
50   POKE 16526, PEEK(X+1): POKE 16527, PEEK(X+2)
60   L=USR(0)
70   X=VARPTR(B$)
80   POKE 16526, PEEK(X+1): POKE 16527, PEEK(X+2)
90   L=USR(0)
100  X=VARPTR(C$)
110  POKE 16526, PEEK(X+1): POKE 16527, PEEK(X+2)
120  L=USR(0)
130  S1=PEEK(28673) + 256 * PEEK(28674)
140  S2=PEEK(28675) + 256 * PEEK(28676)
150  S3=PEEK(28677) + 256 * PEEK(28678)
160  PRINT S1, S2, S3
170  GOTO 40
200  END
```

Sensors such as switches that have only two states (logic 0 or 1) may be connected directly to the inputs of the bus driver and have their state gated directly onto the bus. Of course, the software that reads such a sensor should only test the sensor state once instead of attempting to measure the duration of the signal.

The actual analog to digital conversion is performed by subroutines written in Z-80 assembly language and called by the "USR" command in Level II BASIC. [2] The subroutines, one of which is shown in Table 1, are each embedded in a single line string definition statement in the Level II BASIC program. The machine language code is deposited into the same locations formerly occupied by the characters in the original string statement. This procedure eliminates the need for loading system tapes into protected memory and makes the machine code an integral part of the BASIC program.

To understand how this is done and how the ADC subroutines are used, refer to the BASIC program in Table 2, which reads values from sensors 1, 2, and 3 and prints them. Source statements 10, 20 and 30 are used to store the machine code for the conversion routines for the respective sensors. The address of the subroutine is the byte immediately following the first quote mark of the string definition.

Machine code can either be POKE'd into the string definition locations, or, more conveniently, it can be directly entered into the string location using a monitor program such as RSM or TBUG, residing in high memory (well out of the way of BASIC programs). RSM has the additional advantage that it disassembles the machine code, thus providing a check on the Z-80 opcode mnemonics. The BASIC program should first be entered using arbitrary keyboard characters to fill out the strings (remember to use no fewer characters than the number of bytes in the subroutine plus one zero byte at the end). Then, the machine code in Table 1 can be inserted into the proper string, using the BIT instruction that tests the data bit for the desired sensor. The subroutine shown deposits the result of the conversion into location 7001H. Subroutines for the other sensors may use different locations, such as 7003, 7005, etc. Refer to the footnotes in Table 1 for the code modifications necessary to test other data bits or use different result locations.

Once the machine code is entered, the BASIC program may be stored on tape. The next time it is loaded, the machine code will still be there and the program may be used immediately. The two restrictions on using string definitions to store machine code subroutines are that the subroutine must be no more than 255 bytes and must contain no byte with a value of zero. A zero byte will cause the BASIC interpreter to terminate the string definition prematurely. Careful coding can usually get around both these restrictions. In the ADC subroutine, for example, zeroes in the code are avoided by using 10H as the (arbitrary) port address in the OUT instruction and 7001H, etc., as the result address.

Line 40 in the BASIC program sets the variable X with the address of the ADC subroutine for sensor #1. Line 50 deposits this address in the location used by the USR command, and Line 60 invokes the subroutine. After the conversion is completed, the two byte result is stored in high memory locations 7001 and 7002 (hexadecimal,

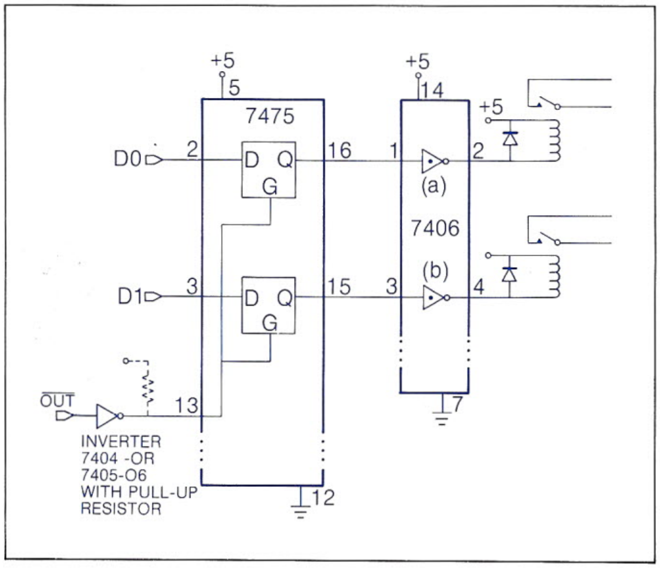


Figure 3. The control signal output interface, which also plugs directly into the TRS-80 expansion port.

corresponding to 28673-4 decimal). Lines 70 through 120 repeat this process for Sensors 2 and 3, whose ADC subroutines deposit their results in consecutively higher locations. Lines 130-150 convert these data to a usable form stored in variables S1, S2, and S3, respectively.

There are other options for the design of the ADC subroutines. Since no output port address decoding is used, the "OUT (10H),A" command in Table 1 will cause all the 555 timer chips to output pulses of duration proportional to their respective sensor voltages. In the subroutine shown, only the desired sensor bit is tested. It would also be possible to configure a routine that looped until the pulse lengths of all timer outputs had been counted. Another alternative would be to write a parameterized routine that could read the value of any one sensor. The number of the desired sensor could either be POKE'd into a parameter location or passed via the USR command linkage. The BASIC program could even POKE the proper instructions directly into the code to access the desired sensor.

In the ADC subroutine given in Table 1, the USR parameter linkage is used to provide an initial value for the loop counter. The "CALL 0A7FH" statement loads the (two byte) value that was given as the argument to the USR statement into registers H and L of the Z-80 CPU. These are then moved into the 16 bit register combination BC which is then incremented each loop cycle. Thus, L=USR(0) initializes the ADC to zero whereas L=USR(100) begins the count at 100. Also, the code shown limits the maximum converter output values to 2047 by exiting when bit 3 of the result becomes a "1." A different limit could be set if desired.

We can now see the great flexibility and ease of implementation achieved by performing part of the A to D function in software. Even more dramatic is the cost savings obtained by this simple method, which requires only a few dollars in parts and for many applications performs as well as multi-channel ADC boards costing hundreds of dollars.

Control of the Robot's Motion

The Kenner R/C robot used in these experiments is controlled by two switches that govern the toy's translation and rotation. For use as a computer-controlled robot, these switches are interfaced to the TRS-80 using the circuit shown in Figure 3. Latched data bits set by the CPU under program control operate relays that in turn key the R/C transmitter.

Upon receiving a positive-going enable pulse, the 7475 4-bit latch chip reads the values on data lines D_0 and D_1 and holds them on the corresponding outputs until the next enable pulse occurs. In this case, the enable pulse is produced as shown by converting the OUT′ pulse produced by the CPU. Each relay is driven by a 7406 open-collector inverter, which grounds one side of the coil when the output of the latch is high (one). When the data bit is low (zero) the coil floats at 5V, leaving the relay inactive. Radio Shack relays #275-004 were chosen because they can be driven directly by the inverter/drivers, needing only 10 milliamps to engage.

The "OUT" command in Level II BASIC has the same function as the Z-80's OUT machine command, placing the desired data on the output bus and generating the OUT′ pulse. In this case an "OUT 0,1" command in a BASIC program will cause the robot to move forward while a "OUT 0,3" command is used to set the direction of the movement. "OUT 0,0" will stop all motion. The amount of the motion is proportional to the duration of the movement—once a motion is initiated by the CPU, it will continue until the CPU generates a new movement command. By using the OUT command and variable time delays it is very easy to program any desired motion sequence in Level II BASIC. [3]

As in the sensor input interface, no I/O port selection logic is used. An OUT command to any port will cause the relays to operate. Thus, the OUT machine code statement used by the ADC subroutines to start the timers could also affect the robot's motion. Since the USR statement used to call the ADCs always sets Z-80 register A to zero, the OUT statement in the ADC code will turn off the relays. If the conversion interval is brief, however, the robot may not have time to respond to this stop command, provided that the controlling program sends out the proper movement command after reading a sensor. Other alternatives would be to read sensor inputs only when the robot is

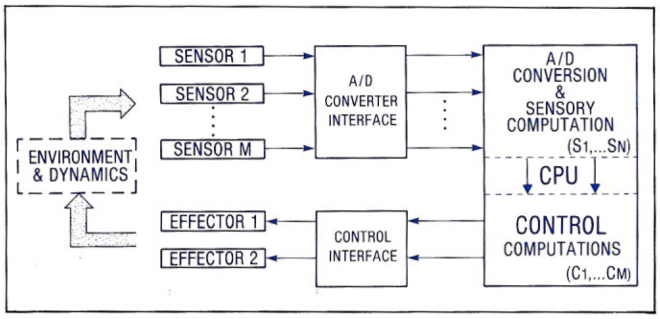


Figure 4. Functional block diagram for the TRS-80 controlled robot.

stopped or to let the same OUT command that controls the robot serve also to start the timers. However, leaving the robot turned on while writing to the cassette or any other peripheral may have unpredictable results!

Investigating Robot Behavior Modes

For a robot with N sensors, the state of all the sensor outputs at a given time can be regarded as an N-dimensional vector:

$$\mathbf{S} = (S_1, S_2, \ldots, S_N)$$

Similarly, if there are M control or actuator states, we can define:

$$\mathbf{C} = (C_1, C_2, \ldots, C_M)$$

to be the M-dimensional control vector.

Complex robot behavior can be defined as the process of computing the feedback control vector **C** as a function of the sensor state vector **S**. The control function can consist of logical tests on the sensor values, as used for the task described below, or else it may involve arithmetic operations on the sensor values or be dependent on the past history of the process. Even with the simple sensory and control capabilities of the inexpensive system described here, complex behavior may be investigated simply by changing and improving the robot control computer program.

With the robot sensory and control interfaces implemented, we can write BASIC programs that will determine a specified robot behavior pattern in response to entered commands and sensory measurements. A functional block diagram for the TRS-80 controlled robot system is shown in Figure 4. The sensors mounted on the robot make measurements on various analog voltage levels. These voltages are converted by the sensor interface, which sends numbers to the control program that are a known function of the sensor measurements.

These data are then processed by the CPU and used as input to a decision-making procedure, which then produces control commands that are sent to the robot by the output interface. In this application, only the two output

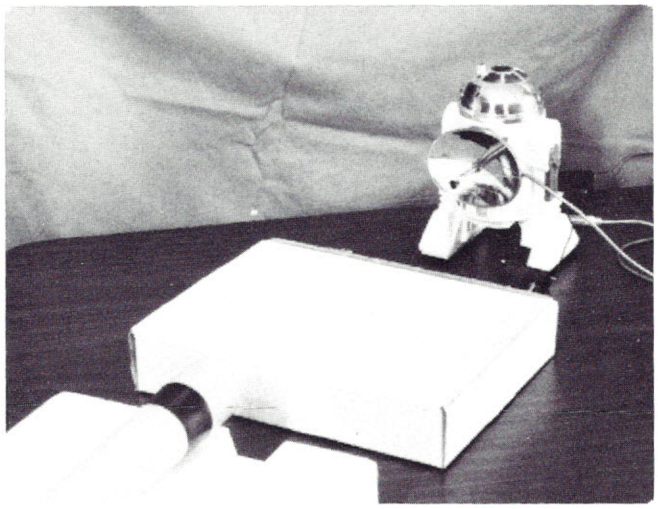


Figure 5. Experimental arrangement for obstacle avoidance behavior.

bits D_0 and D_1 are needed, but this interface can be easily extended (by connecting additional output latches and relays, etc.) to provide up to eight control bits, allowing the program to select one of up to 256 distinct control states by proper choice of the output byte "n" in the BASIC statement "OUT 0,n."

Complex behavior patterns may be implemented in software by coding the desired control responses as a function of the sensor inputs. These responses may just rely on the sensor inputs at a given moment (no sensory memory), or may be based on sensory observations made at earlier moments during the robot's operation.

The process of producing a complex behavior pattern may be illustrated by the following experiment in which the TRS-80 controlled robot detects and tracks an infrared source. In this example, only two sensors were used: the infrared phototransistor scanner and the microswitch touch sensor, both shown earlier in Figure 1. Both the scanner and the touch switch were mounted to the front of the Kenner R2D2, and their signals were returned to the ADC interface by means of lightweight coaxial cable. The R/C transmitter was used for the control signals from the computer output port.

The experiment involves placing the robot at an arbitrary position and orientation relative to the light source, as shown in Figure 5. An obstacle is placed so that it blocks the straight line path between the robot and the light. The robot's task is to detect the light source and go to it, avoiding obstacles as it moves. Breaking this task into primitive operations results in the following procedure:

1. Perform a sequence of IR readings separated by incremental rotations, so that the IR sensor scans the robot's surroundings.
2. Detect the light source, and begin to move toward it.
3. If the touch sensor detects an obstacle, initiate an avoidance maneuver, then repeat from step 1.

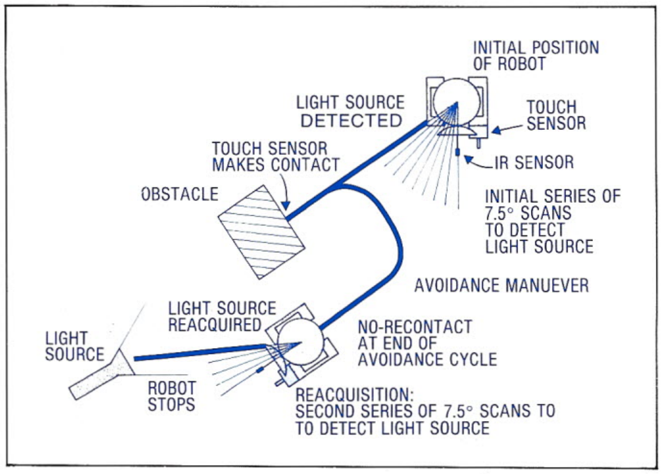


Figure 6. Robot obstacle avoidance behavior. The computer-controlled robot will move toward the light, automatically avoiding an obstruction in its path and stopping when it reaches the light.

4. Continue moving toward light source until the IR reading drops below a threshold, then stop.

In step 1 the robot is rotated in azimuth by 7.5 degree increments. At the end of each increment the IR sensor output is converted and checked to determine if the light has been seen. This is accomplished by testing the IR sensor output against a selectable threshold, using the single BASIC statement:

IF S1>IR THEN GOTO (movement procedure)

The value in S1 may be obtained from a single call to the ADC subroutine, or it may be an average value derived from, say, 10 conversions. Averaging can be used to reduce the likelihood of a false detection if the sensor readings vary greatly due to the presence of electrical or IR noise. The threshold value in variable "IR" is set low enough to give an acceptable probability of detection and high enough to give an acceptably low false alarm rate.

During the forward motion toward the light source, the touch and IR sensor ADC subroutines must be entered periodically by means of the USR command and the resulting values checked. If the touch sensor is found to be on, an appropriate obstacle avoidance routine is called, and repeated until the touch sensor is no longer on at the end of an avoidance cycle. When the robot is very near the light, the collector mirror becomes unfocused, and the IR sensor reading drops below the threshold. A typical ground trace of the robot's motion during this experiment is shown in Figure 6.

Conclusion

Many people make the assumption that robot experimentation is beyond the resources of the average hobbyist. The simple interfaces described here provide the means, both in hardware and software, for interesting experimentation at minimum cost, given that a microcomputer such as the TRS-80, with access to machine code subroutines, is available. The interfaces described here may easily be elaborated by the addition of port selection logic or other hardware, but just as they are they will allow you to begin your own robot experiments with minimum expense and effort.

References

[1] Don Lancaster, "The TTL Cookbook," Howard Sams & Co., 1978.

[2] William Barden, Jr., "TRS-80 Assembly Language Programming," Radio Shack, 1979.

[3] Don McAllister, "Low Cost Robot," *Byte Magazine,* June 1980.

The radio-controlled R2D2 (price appx. $30) can be ordered from: Kenner Products, 2940 Highland Ave., Cincinnati, OH 45212.

GLOSSARY

Complement or logical negation:
Frequently used in digital logic to indicate a signal having the opposite logical value of another named signal. The complement is indicated by an apostrophe following the signal name or a bar over the name. For example, either Q′ or $\bar{Q}$ may be used to represent the complement of signal Q. Thus, when signal Q goes low, for example, Q′ must go high, and vice versa.

ADC:
Analog to Digital Converter—a device for measuring the voltage level of an analog signal by converting it to a corresponding binary number which can be used by a digital computer.

About the Author

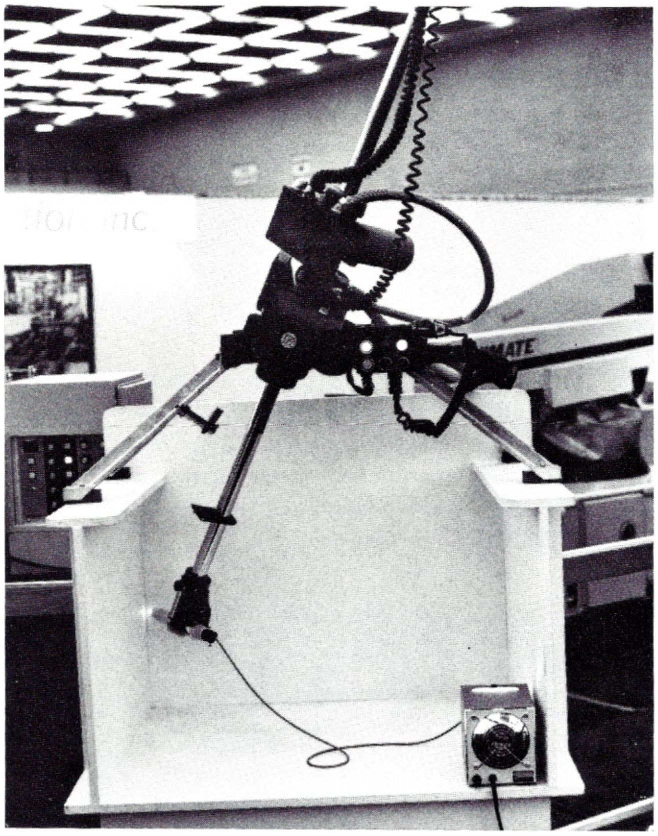
The author, Don McAllister, has built several robot interfaces for his microcomputer. He has worked in the areas of guidance and control as applied to the Apollo and Minuteman missile programs and is currently involved in sensor data processing at the Aerospace Corporation. He obtained his B.S. in Physics at the California Institute of Technology and his M.S. in Astronautics at the Massachusetts Institute of Technology.

The Robots of Autofact II

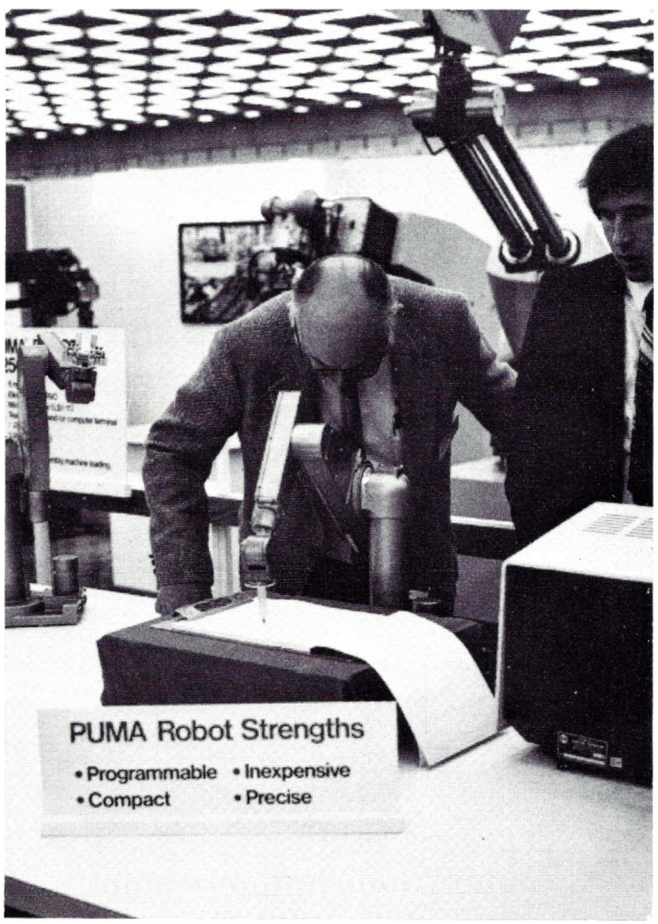
"Toward the Automated, Integrated Factory of Tomorrow ..."

The Society of Manufacturing Engineers (SME) held its second conference on the Automated Factory, AUTOFACT II, last Fall at Detroit's Cobo Hall. Over 7,000 people attended the seminars and viewed displays and demonstrations by most manufacturers of robotics-related equipment. AUTOFACT II comprised six concurrent conferences: Assemblex VI—aimed at cutting assembly cost through automation, CAD/CAM VII—on Computer-Aided Design and Manufacturing, Materials Flow I—dealing with the logistics problems common to mass production, Predictive Maintenance I, Qualinspex I—on Quality Control methods, and, of course, Robots IV.

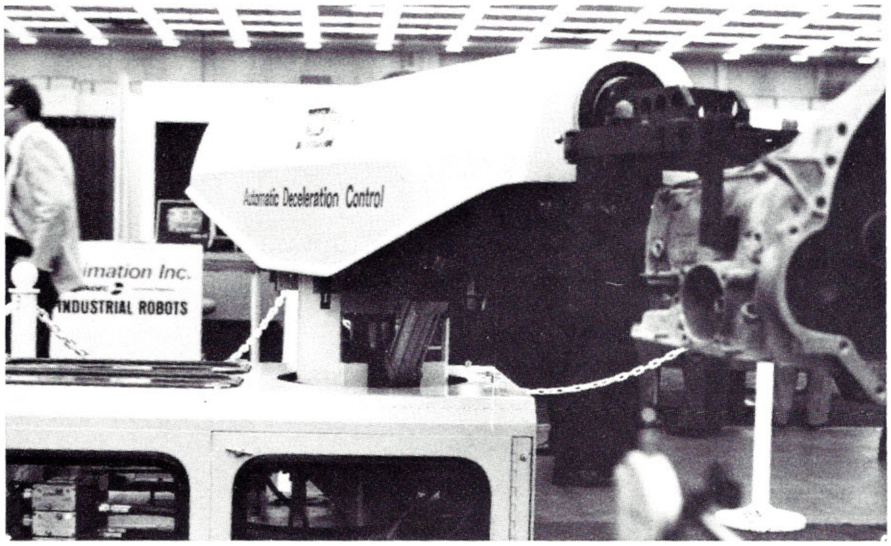
The AUTOFACT conferences are designed primarily to inform industrial management of the latest developments in the fields covered. Many of the systems featured at AUTOFACT II were described or demonstrated earlier at the more research-oriented ISIR9. *(See Industrial Robotics '79, in Robotics Age, Summer '79.)* Nonetheless, several of the systems shown here were unveiled at AUTOFACT II.



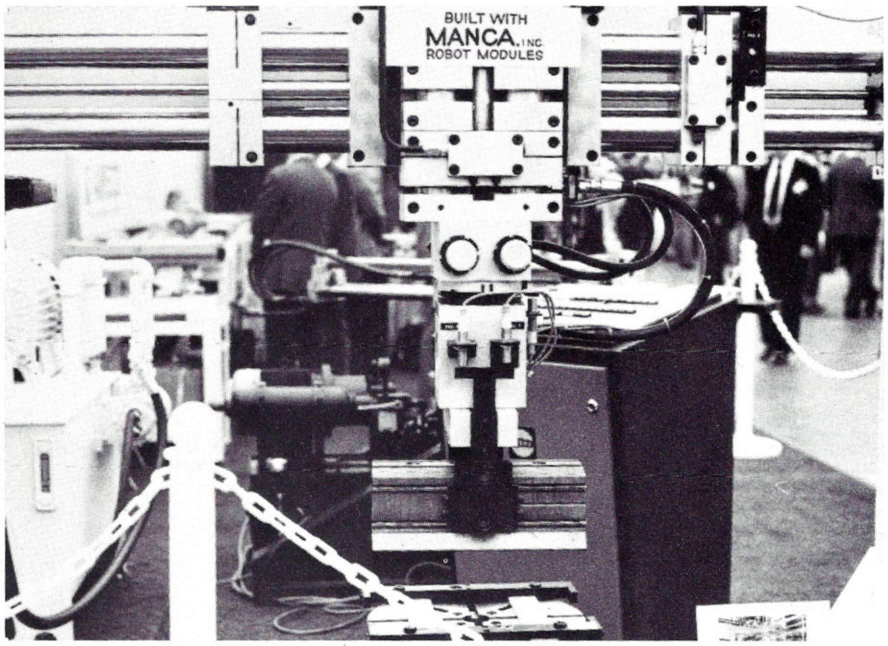
Unimation's "Apprentice" is a portable robot designed for welding. The robot is demonstrating a simulated weld pattern using a flashlight beam. The "Apprentice" is taught the path of the weld using a teaching head. The head is slipped over the welding unit and then moved along the seam manually by the operator as the robot records the path. This rapid learning ability allows one operator to position and operate several "Apprentices."



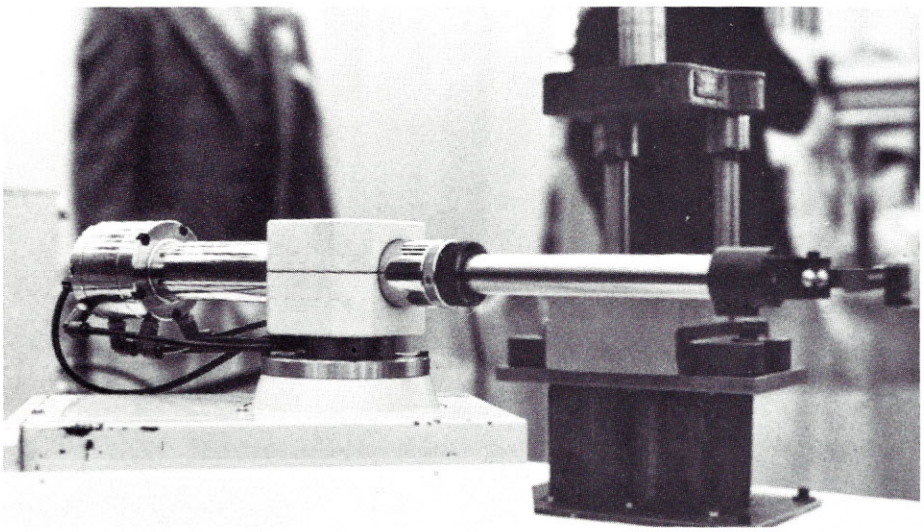
Unimation Inc. introduced its new smaller version of the Puma, the Series 250. With a load capacity of 3.3 lbs. and 5 ft/sec speed, Unimation is aiming this robot at small appliance and electronics assembly applications.



Prab Conveyors demonstrated the high payload capacity of one of their larger robots by having it maneuver a heavy transmission casing from point to point with high accuracy.



The MANCA Inc. modular parts handling system is shown hefting a large weight. MANCA manufactures two lengths of hydraulic linear translators, a rotary (wrist) module, and two types of grippers. These units can be combined as desired, and each type is available in four sizes.

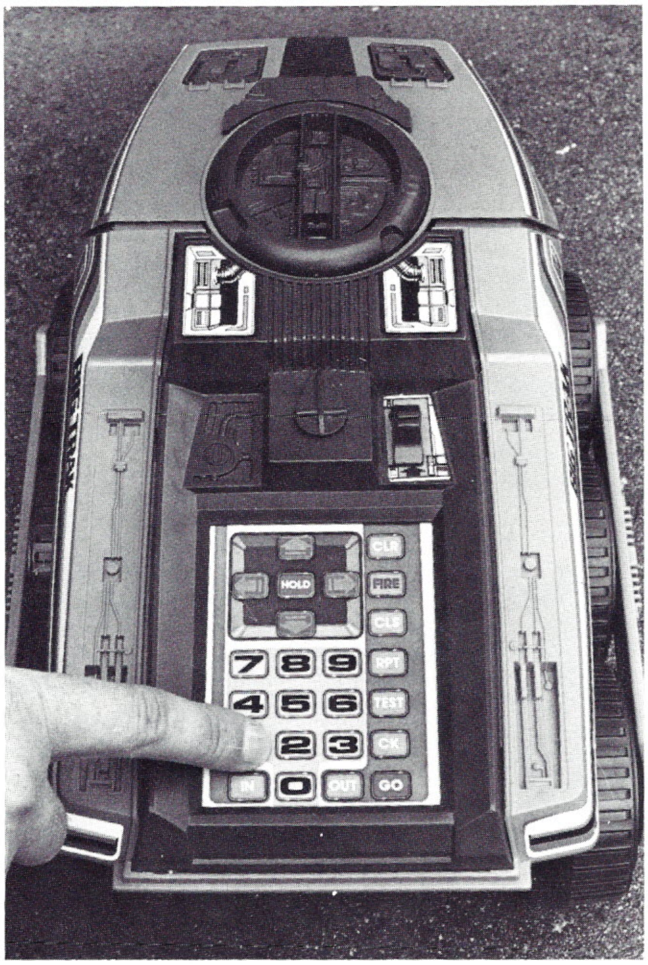


Seiko's Model 700 made its first appearance in the U.S. at AUTOFACT. This portable unit has been used for many years by Seiko in its own factories. In this demonstration the 700 grasped a component from a delivery chute and positioned it at another location.

A view of General Electric's TN 2500 CID (Charge Injection Device) Camera. With the lens removed (above) the CID imaging chip is visible. The camera was demonstrated in conjunction with GE's new PN 2150 TV Frame Buffer Device, which provides a means of capturing and storing a single frame of video information (see New Products, Winter 1979).

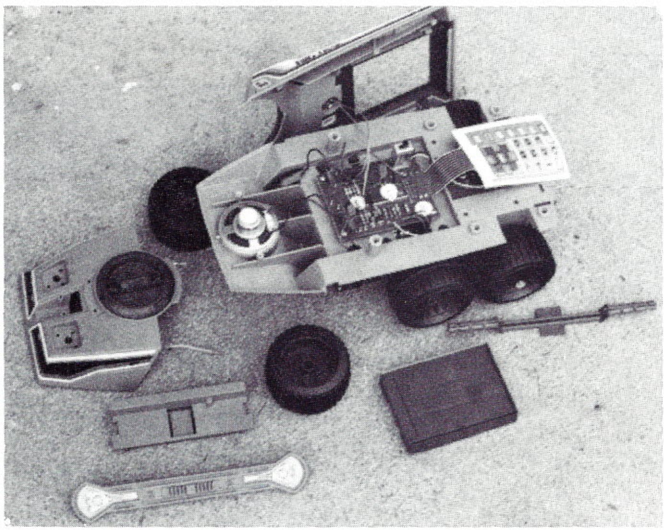
Object Recognition System's unique display demonstrated the ability of their vision system to distinguish between various similarly shaped products and parts, as they moved past its camera on a model train. Note the product name displayed in the upper left corner of the monitor.

INSIDE "BIG TRAK"*



New computerized toys may provide a source for cheap robot hardware.

When we heard about Milton Bradley's new "Big Trak™" programmable toy tank, we wondered how they were able to include a microprocessor controller and still retail the toy for so little. (Our local toy shops have it for about $40.00.) To find out, we contacted Milton Bradley and spoke with Mel Taft, Milton Bradley's Senior Vice President of R & D, who kindly offered us a Big Trak, and gave us a few facts about it and information on his company's plans for future computer-controlled toys.



The tank is programmed by a paper-thin membrane keyboard (see photos), allowing the user to enter a sequence of up to 16 different commands. Most commands take a two-digit argument that is interpreted differently depending upon the command, specifying the amount of time, distance, rotation, etc. Apart from the motor drives, two program-controlled outputs are available, one that activates an accessory port and another that drives a lightbulb, coincident with sound effects for the tank's "phasers." A dump-truck trailer is available that uses the accessory port, and other accessories are planned.

The toy's controller is TI's TMS 1000 4-bit microcomputer-on-a-chip. The 1024 instruction device includes an ALU, ROM, and I/O in a 28-pin package. The TMS 1000 scans the keypad directly, composes the sound effects that accompany the tank's operation, and, using the only other IC in the unit, a 75494 output driver, sends its orders to the tank.

High volume production, of course, is the key to the

*"Big Trak" is a registered trademark of Milton Bradley, Inc.

toy's relatively low cost. Milton Bradley expects to market over one million of the tanks this year, according to Taft.

The major toy manufacturers are convinced that computerized toys will sustain their already great popularity, and continued development of innovative products is vital. Mr. Taft told Robotics Age that Milton Bradley is currently spending over $600,000 per year for research on robot- and computer-related products. *(This is roughly comparable to the entire NASA robotics research budget—ed.)*

One of the products of this effort is a proprietary voice synthesis chip which MB expects to begin using later this year. The chip can store in ROM about 22 different words and will cost the manufacturer a little over $3.00. Additional 22 word increments may be added at a cost of about $1.65 per ROM chip. Using these, future models of Big Trak will talk back to their owners.

But this is just the beginning, according to Taft. Eventually, MB toys will feature sonic and contact sensing, and even a robot manipulator is in the works.

All this should be important news to Robotics Age readers. As the toy industry, already past masters at producing inexpensive mechanisms, discovers the vast potentials of robotics, we can expect to see an increasing array of cheap, practical devices suitable for computer interfacing, providing an attractive alternative to time-consuming hand-built items and expensive commercial equipment.

One final note—Mr. Taft tells us that Milton Bradley licenses the designs of about half of the devices it uses in its toys from outside sources, so you may want to send them that unique new sensor you've built!

MEDIA SENSORS

Early this year, ROBOTICS AGE was contacted by the producers of the Merv Griffin Show to help them plan a segment about robots for one of their programs. It was not a very easy request to satisfy. We suggested that the segment attempt to give a realistic impression of the current state of robot technology and what the near future might bring, but there was also the requirement that the show be entertaining and understandable to a TV audience.

Even so, they liked our suggestions, so we went on to propose demonstrating a state-of-the-art industrial robot, as well as an inexpensive hobbyist unit. Having just visited AUTOFACT II, where Unimation unveiled its new Series 250 minature PUMA-type arm, we immediately contacted the Danbury firm about having one on the show. We also approached Lour Control, makers of a hobbyist robot kit. Fortunately, both agreed to participate, and we were pleased that Joe Engleberger, founder of Unimation and often described as the "father of industrial robotics," consented to appear as a guest on the show.

With the addition of an Apple computer, RF link interface and voice recognition board provided by ComputerWorld of Van Nuys and Heuristics, Inc., the Lour robot shell was made to respond to spoken movement commands. The robot performed well during rehearsals, but during the actual taping the RF noise from the spotlights and wireless mikes on stage resulted in erratic, but acceptable, control.

The star of the show was the Unimate manipulator. Engineers at Unimation West had prepared an elaborate demo of the little arm's capabilities. Brian Carlisle, General Manager of Unimation West, and Bruce Shimano, developer of Unimation's VAL robot programming language, were on hand to set up the system. The robot was mounted in front of a closed, curtained window. After reaching around the curtain and pulling the cord to open it, the robot picked up a window cleaner and sponged the lower pane. Turning the tool over to the rubber blade on the other side, it then proceeded to "squeegee" the window dry.

The demo didn't stop there. After replacing the tool, the robot unfastened the window latch and raised the window. It then completed its performance by watering each of the flowers in a window box on the other side of the frame. Mr. Engelberger did an excellent job of relating the robot's talents to the audience. Although the segment was too brief to elaborate on any of the issues, Engelberger suggested that such robots would eventually be available as versatile and uncomplaining housekeepers. The show aired early in March in most major cities.

Automatix Inc. Begins Operations—*Wall St. Journal,* Jan. 31, 1980. A new company with impressive credentials and an

appropriate monogram is to begin operations in February, entering the fields of programmable automation and vision-based industrial robotics.

Automatix Inc.—whose initials also stand for Artificial Intelligence—will manufacture and market industrial robots and related products.

The company, in announcing its formation, said it expects to be the first manufacturer to supply modular turnkey robotics systems for industrial applications. (Turnkey systems can perform in varying applications, meeting needs common to many different users, instead of being custom built to fit one company's specific needs.)

Robots by Automatix will feature television cameras, enabling them to use visual information in their functioning. The first AI products will include a robot arc welder and sorting and assembly machines.

Automatix begins its operations with strong financial backing and a nine-person management team with an impressive background.

Said President Philippe Villers, in an interview with the Concord Journal (Jan. 24, 1980), "We're financially strong as a result of our investors," who have put "just under $6 million" into the fledgling operation. The investors, reported the Journal, include Harvard University and the Massachusetts Institute of Technology.

Villers will attempt for the second time to start a pioneering computer firm. His first success began in 1968, when he co-founded Computervision Corp., the first large-scale manufacturer of computer-aided design systems. He served as that company's senior vice president until resigning to co-found Automatix.

His co-founders are:

John Dias, vice president for finance, who held the same position at Data General Corp. He worked there during its growth years, 1969-79.

Mike Cronin, vice president for marketing, who was a Computervision executive for nearly 10 years, most recently as executive vice president of the productivity systems division.

Donald Pieper, vice president for research and development, who was general manager of technical operations at Continental Can Corp., in charge of computer-aided manufacturing technology.

Victor Scheinman, vice president for advanced systems, former general manager of Unimation West. Scheinman, who Fortune magazine called "father of the new robots," invented the Vicarm and Puma robots.

Daniel Nigro, vice president for manufacturing, who was with Data General for 11 years, most recently as director of international manufacturing.

Gordon Vanderbrug, product-line manager, who headed the widely acclaimed vision-based robotics program at the National Bureau of Standards.

Arnold Reinhold, product-line manager, who was director for advanced applications at Computervision, managing software developmennt groups.

Norman Wittels, product-line manager, formerly a research manager at the Sperry Corporate Research Center.

Automatix is developing its first three robot systems: Robovision, a vision-based robot arc welder; Cybervision, an intelligent programmable assembly system; and Autopose, which does sorting, inspection and positioning.

Robovision, said Villers, will be able to weld anything, including earth-moving machinery, tractors and trailers. Cybervision and Autopose will have "flexible automation" capabilities, he said, as their computer bases will permit them to be programmed for numerous functions.

Computer Speaks Plain English

Help may be on the way for those of us who would like to "speak" to computers but don't have the necessary language skills, according to an article in *Popular Science* (February 1980).

It soon may be possible to address a computer in English, the article reports, thanks to a Philips system now in the early development and testing stages.

The Philips Question Answering system could eventually open up organizational computers to far wider use, giving anyone who speaks English access to computer data banks. Furthermore, early PHILQA work "points to exciting future developments for home units."

With PHILQA, questions can be composed of any choice of words or sentence structure. The computer attempts to answer the question by breaking it down in three steps:

1) Establishing the grammatical structure of the sentence and the logical relationships between sentence parts, by referring to an English grammar stored in the primary memory.

2) Identifying nouns and verbs, by referring to a data bank with a selected vocabulary related to the particular topic in question.

3) Translating the question into data-base language and sending it to the main data bank in this form.

The computer displays its answer on a CRT screen. "If the reply is

ambiguous," the article explains, "the computer will search for another interpretation, and the whole process is repeated."

Robot Transmission Shifter It can't quite drive a car, but it can "literally *shift* for itself, operating clutch, accelerator and stick shift," reports *Popular Science* (February 1980) in its *Science Newsfront* section.

The device referred to is a robot shifter that is being used during 50,000-mile durability measurement on test-cell dynamometers at Chrysler's Chelsea, Michigan, proving grounds.

Chrysler engineers believe that their invention is the first successful robot shifter for manual transmissions. Such robot testing has been done previously, but only on automatic-transmission cars.

A microcomputer acts as both programmer and controller, instructing the robot as it monitors and controls the throttle, clutch and shift lever.

"Those Smart Young Robots on the Production Line" "An army of small and relatively inexpensive robots *is* taking over more and more jobs that were previously performed by humans," writes Gene Bylinsky in an informative article appearing in *Fortune* (Dec. 17, 1980).

Microelectronic technology, the writer says, has made the new breed of robots "sylphlike" and "far brainier," in stark contrast to the "hulking brutes" first introduced to the industrial scene in the early 1960s. Today's computers enable robots to learn "a succession of tasks," with "a versatility that promises to render obsolete a good deal of what is currently thought of as automation."

As robots continue to get smarter, developing a sense of "touch" as well as "sight", we move closer to the day when they could take over many of the functions by which manufactured goods are produced in this country.

Bylinsky examines some of the ramifications of this trend and scans the robotics field for some of the more interesting innovations and their creators.

"Contrary to general belief," he writes, "75 percent of U.S. industry's products are assembled in small batches as styles and sizes change." Robots could take over many such batch-assembly jobs, today done largely by hand. Thus, the potential exists for widespread applications of robot technology.

That potential is being realized by four companies at the forefront of the young industry, which commands a $60 million market in the United States:

Unimation Inc. of Danbury, CT., which is the world's largest robot manufacturer and has more than half the U.S. market; Prab Conveyors Inc. of Kalamazoo, Mich., Auto-Place Inc. of Troy, Mich.; and Cincinnati Milacron's robot division. All report soaring demand for robots.

"The number of industrial robots in use in the United States has more than doubled to 3,000 in the past seven years," the writer notes. "For the past three years, the industry's sales have been growing at an annual rate of 35 percent." Ten years from now, U.S. sales are expected to total $700 million to $2 billion.

These numbers are attracting the interest of some of the "big guys," too. Texas Instruments is exploring the possibility of marketing the robots it now makes for its own use. "IBM is, too," Bylinsky says, "without admitting it."

Other attractive figures: It costs no more to maintain a typical robot today than it did in the early 1960s—about $4.60 and hour. Relatively speaking, then, robot "labor" has grown cheaper over the years. In one typical manufacturing application, robots reduced labor requirements by 70 percent, increased production by 10 percent, and cut rejects by 15 percent.

In conclusion, writer Bylinsky poses this question: "Does all this mean that robots are on the verge of displacing people en masse?" Two observers provide answers.

James Albus of the National Bureau of Standards believes that "in a few years, we can have factories where workers are primarily involved in supervising robots." If the growth of robotics can parallel that of computers, he says, "we would see robots making robots, bringing machines costing thousands now down to less than $100 each. This could mean an effective labor rate, in many operations, of only pennies per hour."

A more sober view held by many current robot users, is that most jobs would still be performed by people. After all, says Paul F. Gray of Ford Motors, "The robot is up against some pretty stiff competition—the human being." Even J. F. Engelberger, founder of thriving Unimation Inc., estimates that even at the highest projected growth rate robots will replace no more than 5 percent of the Western world's blue-collar work force by the end of the century.

ORGANIZATIONS

The International Robotics Foundation

A new organization has been formed to promote the development of low-cost, general-purpose robots. The non-profit International Robotics Foundation (IRF) has attracted considerable attention by offering a $50,000 prize for the first robot that can successfully perform each of a carefully selected set of tasks. The tests, to be designed by a panel of robotics experts, include many housework-related chores. Also, the foundation will be responsible for the administration of the Robotics Age Competitive Event (RACE) announced in Vol. 1 No. 1 of this magazine. The latter contest will still be sponsored by ROBOTICS AGE, but the IRF will be responsible for developing the rules and organizing the event.

The Foundation will offer numerous services to its members. A regular newsletter will report on IRF acitivities, including the specifications of the RACE rules and the IRF Prize criteria. An information service for members will offer prompt response to questions. Members can also subscribe to ROBOTICS AGE at a 15% discount. IRF Membership dues are $10.00 per year. For more information contact: International Robotics Foundation, 3011 Community Ave., La Crescenta, CA 91214.

Society of Manufacturing Engineers

The Society of Manufacturing Engineers recently announced its plans for two follow-ups to 1979's Autofact II, the latest SME automation conference and exposition.

Autofact West will be held Nov. 18-20 at the Anaheim, California Convention Center; Autofact III will be held Nov. 10-12, 1981, at Cobo Hall in Detroit.

The topic of Autofact II, "The Automated, Integrated Factory," drew more than 7,000 people to the three-day exposition, Oct. 30-Nov. 1, at Cobo Hall. Nearly 1,700 registered for the conference.

Bill Hilty, SME managing director for expositions, predicts equally strong interest in this year's Autofact West.

Said Hilty: "Based on the growth of computer automation in industry, there seems to be little doubt that Autofact will be SME's leading vertical-subject conference and exposition.

"Autofact West will be another step forward, given the high concentration of aerospace, electronics and industrial diversification in Southern California. Detroit served as the proving ground for the discussions and demonstrations of automated factory techniques. The West Coast is another area where Automation is gaining applications."

Autofact West will focus on Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM)

(continued page 51)

Calendar of Events

Symposium in Cognitive Science

The Vassar College Cognitive Science Group will hold a two-day symposium, April 25 and 26, 1980 at the Vassar campus in Poughkeepsie, NY. The symposium will be devoted to an exploration of the role which context plays in the perception and interpretation of language. Context will be considered from the points of view of social, perceptual, intentional, linguistic, and computational analysis and the ways in which they are related.

Guest speakers, representing a broad range of disciplines, will include: Roger C. Schank, Artificial Intelligence, Yale University. Daniel C. Dennett, Philosophy, Tufts University. James D. McCawley, Linguistics, University of Chicago. Howard Gardner, Psychology, Harvard University. Jerre Levy, Biopsychology, University of Chicago.

The symposium will begin at 1 pm on Friday and will conclude at 3:30 pm on Saturday. A dinner for speakers and attendees will be held Friday evening.

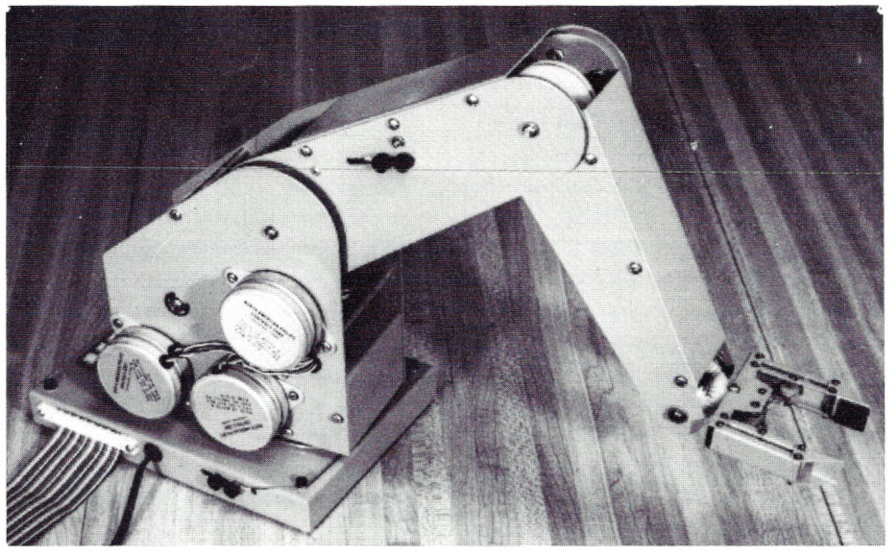
Participation in the conference will be limited to the first 150 people who register and pre-registration is necessary. To obtain further information and/or registration materials, please contact: Cognitive Science Symposium, Vassar College, Box 525, Poughkeepsie, NY 12601. (914)452-7000, ext. 2407.

National Artificial Intelligence Conference, August 19-21, 1980. Stanford University, Palo Alto, CA. (see announcement in Vol. 1 No. 2)

1980 LISP Conference, August 24-27, 1980. Stanford University. (see announcement in Vol 1. No. 2).

NEW PRODUCTS

The MiniMover 5 by Microbot



The MiniMover 5 tabletop arm is a unique instrument that attaches as a manipulative device to an inexpensive personal computer. It enables individuals or groups—such as schools and technical-interest clubs—to acquire "hands-on" experience with computer controlled automation, artificial intelligence, and robotics. The MiniMover 5 may be used for such applications as: 1) computer games, in which the arm moves game pieces on command; 2) computerized construction, in which building components may be arranged into a wide variety of configurations or mathematically programmed designs; 3) computer assembly, simulating automated factories of the future; 4) computer art, using graphic instruments such as paint brushes, felt tip pens, etc. directly.

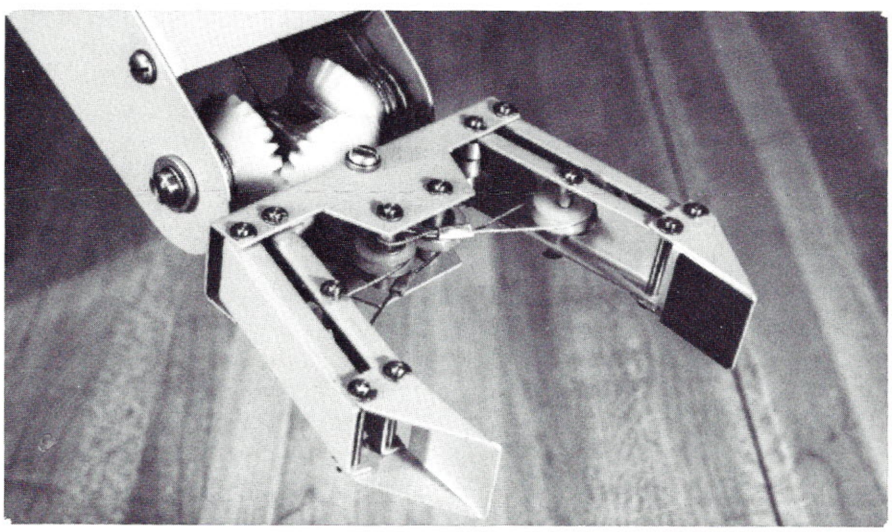
A complete hardware and software package has been developed to run the MiniMover 5/80 version with the Radio Shack TRS-80 Computer (Level II). The hardware consists of the arm, its power unit, and a ribbon cable connection to the TRS-80 keyboard. For interfacing with other computers, the MiniMover 5/8P version is controlled by a single 8-bit parallel port.

The ARMBASIC software package allows control of the MiniMover and its hand by simple BASIC-like commands. The assembly language motor drivers and the Cartesian coordinate transformations are included. Sample applications programs for calibration and block construction are available.

The MiniMover 5 is a five-jointed arm with a lifting capacity of 8 oz. when fully extended. Controlled by stepping motors, it can position with an accuracy of 0.013 inch. The parallel-jaw hand grasps objects 0-3 inches wide and may be positioned inside a partial sphere with a radius of 17.5 inches. Top speed is from 2 to 12 inches per second depending on the weight of the object being handled.

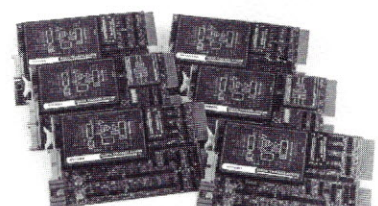
Further information on the MiniMover 5 can be obtained from Microbot, 1259 El Camino Real, Ste 200, Menlo Park, CA 94025.

Circle 8



Analog Interface Boards from DTI

Data Translation Inc. has two new models of high-resolution,



single-board analog interface systems.

The new systems, adjuncts to DTI's standard 12-bit systems, have 14- and 16-bit analog-to-digital conversion, respectively, and backplane and protocol compatibility with the DEC LSI-11/2 microcomputer.

Mounted on DEC-standard dual height cards, the 14- and 16-bit models have fully expandable multiplexer inputs available with either 16 single-ended or 8 differential analog channels.

Both high-resolution models are available with an optional software-programmable gain amplifier. The high-input systems use the PGH option for gains of 1, 2, 4 and 8; low-level-input systems use the PGL option for digitally selectable gains of 1, 10, 100, and 500.

The combination of enhanced resolution and software-programmable gain enables the systems to accurately resolve, measure and confirm the values of extremely small signals that otherwise would be masked by ambient system noise. Thermocouple measurement, for example, is one application that requires this high degree of resolving power.

Models range in price from $625 to $2,070, depending on choice of resolution, throughput and options.

For more-detailed information, write or call Data Translation Inc., 4 Strathmore Rd., Natick, MA 01760. 617/655-5300.

Circle 9

Services for Robot Users

Robotics Technology, Inc., near Atlanta, is a company whose major objective is to help accelerate the use of robots in the United States by providing services to current and potential users.

The company describes itself as an "objective, knowledgeable" source of currently available robotics technology. RTI is an "effective" engineering firm applying robots to manufacturing operations throughout the country. Its activities include:

- Intensive education of robot users through technical programs, often in industries and geographical areas that previously received minimal, if any, exposure to the field.
- In-plant analysis of potential applications, drawing on experience with hundreds of applications for small companies and large corporations.
- Applications engineering systems fabrication and implementation, objectively using the most effective robots for a given operation.
- Creation of the first permanent robot-demonstration facility, where all commercially available models will be on display all year long, each fully operational for use in demonstrations and trials for potential applications.

With its engineering and technical facilities, RTI can mock up and physically evaluate potential applications that may require too much floor space and development time for the robot manufacturer to provide.

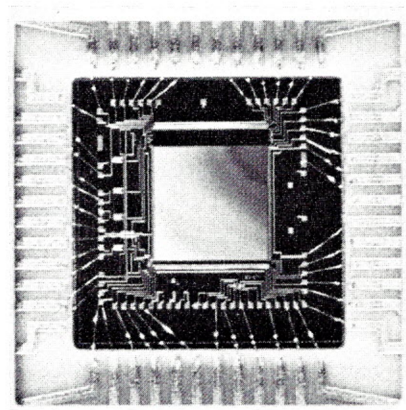
The company also has established a comprehensive audio-visual and technical-data library for use in public education sessions across the country, in-plant training programs and demonstrations at RTI headquarters.

When the company's permanent display of fully operational robots opens this summer, the potential user can experience the various commercially available robots during one trip to the Atlanta area any time of the year.

Write or call for more information: Robotics Technology Inc., 5 Technology Park/Atlanta, Norcross, GA 30092. 404/447-9310.

Circle 10

Charge-Coupled Imager



Hughes Omneye imagers comprise a special series of charge-coupled devices designed with commercial and industrial uses in mind.

These compact imagers are suited for use in sizing, orienting, identification and numerous other industrial control jobs.

The HCCI 100A array can be operated in either of two modes: as an imager, much like a camera, or as a line scanner with 100 time-delay and integration stages. The line scanner is used where images are moving and when the greatest response with the lowest possible noise is necessary.

The CCI output is a series of pulses, the height of each being the analog of the light incident on the associated picture element (pixel). An analog-to-digital circuit converts the image into data compatible with digital computers.

Hughes CCIs offer these advantages over vidicons (which employ photoconductive camera tubes): precise, accurate and repeatable spatial information; the reliability

and ruggedness inherent to solid-state devices; low voltage and power, simple interface with digital data-processing equipment; easy matching to charge-coupled device memories.

In addition, special functions can be designed easily into the Hughes CCI devices.

For more details, write or call Hughes Aircraft Co., Industrial Products Div., Image and Display Products, 6155 El Camino Real, Carlsbad, CA 92008. 714/438-9191.

Circle 11

Computer Interface from CmC

The Connecticutt microComputer μDAC system now includes an interface to the BSR X-10 remote-control module, which makes possible computer control of lamps, motors and appliances.

The CmC X-10 interface enables a computer to control up to 256 separate devices at once: lamps (which can be turned off and on, dimmed, brightened), alarms, kitchen appliances, stereos, TV sets, motors, pumps, heaters and more. The system features plug-in compatibility and software for most microcomputers.

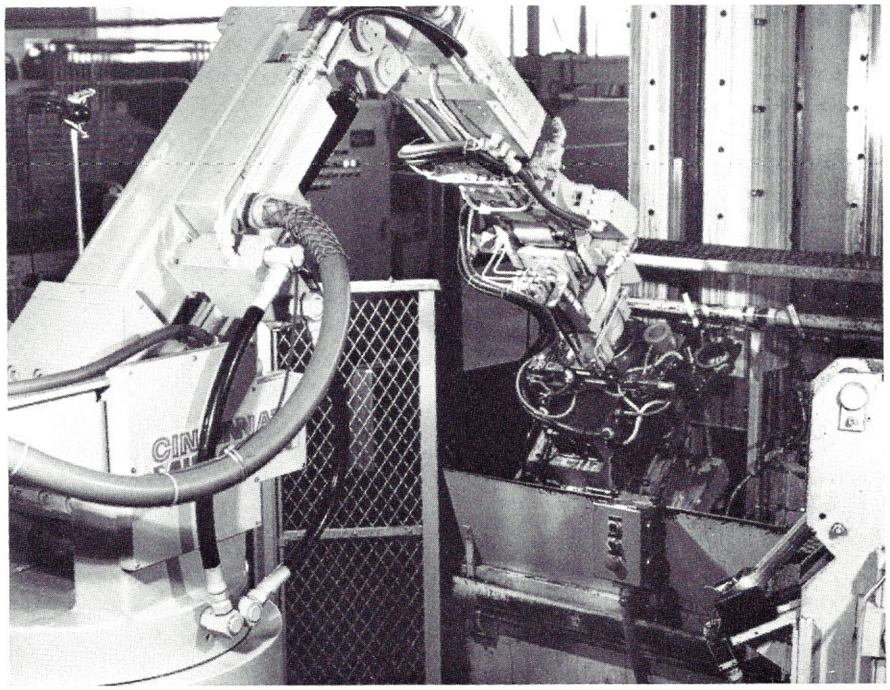
Further details are available from Connecticutt microComputer Inc., 150 Pocono Rd., Brookfield, CT 06804. 203/775-9659.

Circle 12

Another Application for T³

Cincinnati Milacron's T³ industrial robots continue to find applications with U.S. manufacturing firms, including Cummins Engine Co. of Columbus, Ind.

Cummins uses its T³, equipped with a double gripper, to load and



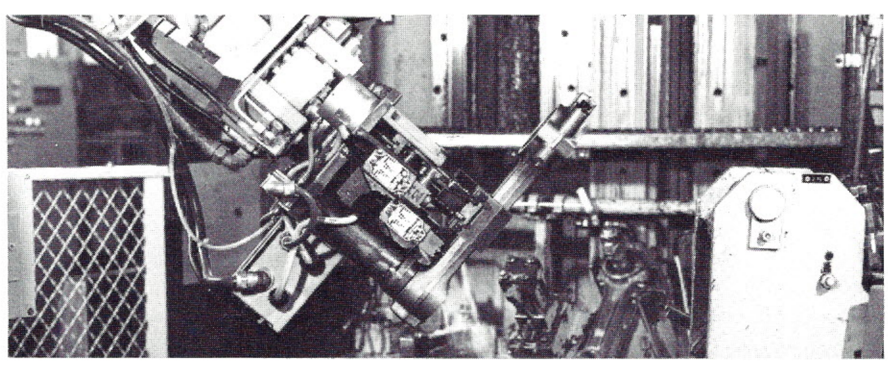
unload a duplex vertical broaching machine that produces finished steel connecting rods.

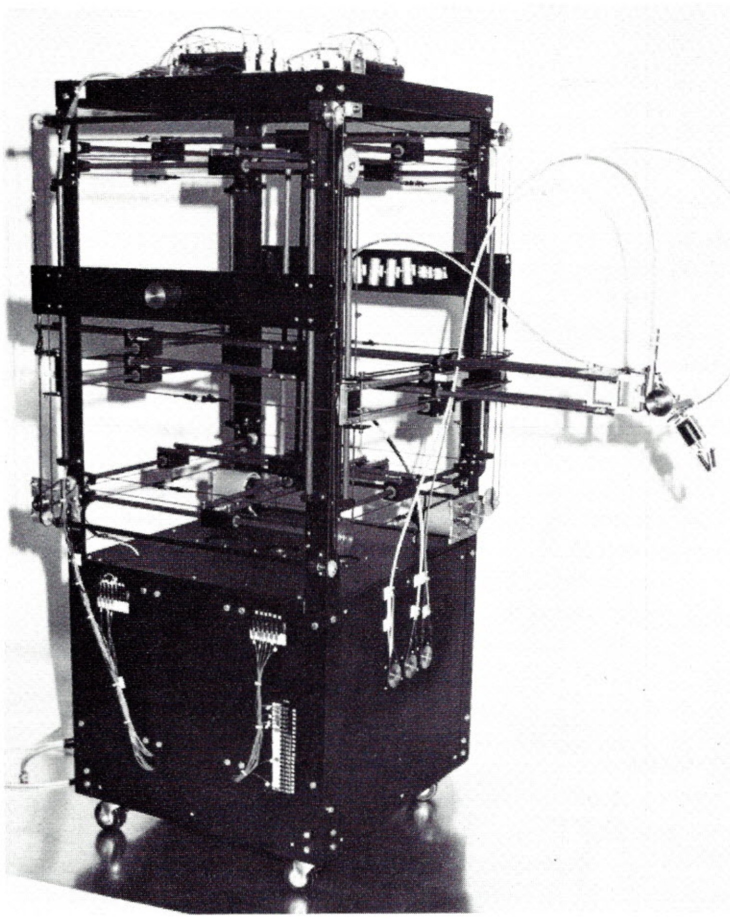
The robot takes an unfinished rod from an incoming conveyor and interchanges it with a machined part in the broach fixture. The robot's computer-based Acramatic control is interfaced to the broaching machine, completely regulating the automatic cycle. The robot-machine system produces a finished rod every 13½ seconds.

The hand-coordinated motion of the T³ moves the arm through as many as all six available axes (x, y, z, roll, pitch, yaw) simultaneously. Because all axes are coordinated, the arm moves in a single straight line from point to point and maintains hand orientation, even though as many as six axes are in motion.

Such straight-line motion is essential to place the pin end of the connecting rod on the tapered locator in the broach fixture, since the clearance between the locator and the connecting rod is only 0.004 inches.

For more information about the T³, contact Cincinnati Milacron, Cincinnati, OH 45209 Circle 13





New GRIVET Series 4 Robot

The Gallaher Research Inc. GRIVET Series 4 Model 1 industrial robot can generate straight-line motion in any direction along any plane and perform non-linear tasks as well.

This unique versatility enables the Model 1 to adapt itself to the work site with only minimum alteration to existing equipment. The robot can handle objects weighing three pounds or less for packaging and assembly purposes. Possible future models may be able to handle loads up to 50 pounds.

Using timing belt drives and digitally encoded stepping motors, the Model 1 can position its hand with high speed and accuracy at any designated point within its work envelope. Adding a three-axis end-effector attachment to the arm boom provides yaw, pitch and roll motions, aiding in the final orientation of the work piece. A solenoid-actuated gripper with a one-inch grasp is provided with the Model 1 evaluation kit.

The Model 1 is completely self-contained, with fan-cooled power supplies, 10-slot card cage, microcomputer-controller, mini-floppy disk drive and air filter housed inside the base module.

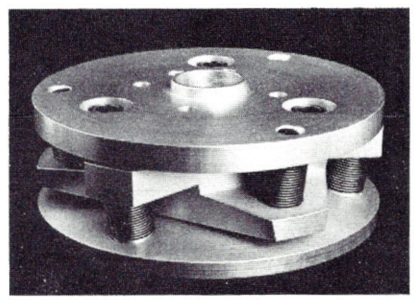
On-board firmware, among its other functions, performs executive scheduling, provides remote data communication and contains a diagnostic debug monitor.

Thirty-two parallel input/output lines allow the computer to interface itself with its environment—not just respond to it—and control every aspect of the process involved. Additional interface and memory boards may be placed inside the system to enhance performance and provide analog-to-digital and digital-to-analog translations. The robot would then be able to make decisions based on changes in temperature, humidity, pressure or conveyor-belt speed.

The Model 1's RS232C communications port gives the user remote access to the system for information processing, inventory control or material-flow analysis. Several robots linked together through this port would comprise a distributed robot system. A master robot could command up to 32 "slaves."

Direct inquiries for more information to Gallaher Research Inc., P. O. Box 10767, Winston-Salem, NC 27108. 919/748-8761. Circle 14

Passive Compliance for Automatic Assembly



The Astek Accomodator(TM) provides engineered multi-axis compliance or float for automatic assembly machines. The use of compliance permits precision insertion operations at closer tolerances than the positioning accuracy of the machine. Designed around high-performance elastomeric shear pads, the Accomodator(TM) compensates for both position and out-of-square alignment errors, minimizing the assembly forces and the probability of parts jamming.

Specifically designed for use on small robotic assembly machines the light-weight ASP-100 Accomodator(TM) mounts directly on the Unimation PUMA robot and is available with interfaces for the

standard PUMA tooling or the ASTEK gripper. Other robot and special tooling interfaces can be provided. Contact ASTEK Engineering Inc., 5 Bridge St., P. O. Box 7201, Watertown, MA 02172. 617/924-2929. Circle 15

New Robotics Newsletter

A new monthly newsletter will soon be tracking the trends in the developing field of industrial robotics.

The new publication is *Industrial Robots*, from Technical Insights Inc. Publisher Kenneth A. Kovaly says his newsletter will be keeping readers abreast of applications, developments in industry, and the rapidly changing technology that is emerging from research laboratories and robot manufacturers.

"U.S. industry," Kovaly says, "is on the brink of a fundamental change in the way materials are handled in factories, and in the way products are assembled and tools manipulated." Certain pioneering companies, he notes, are already using robots to contribute dramatically to productivity.

The newsletter will focus on certain robots: those programmable articulated arms that are adaptable to changing products and models.

To understand the great potential that exists for robots in manufacturing, one must take note of a surprising fact: 75 percent of all U.S. industrial production is based not on long mass-production lines but on batch assembly. That is, most production runs number as few as 50 items for any one product style. At this level of output, Kovaly says, no company can install fixed automatic handling equipment. But it can install programmable robots, whose sequence of movements can be changed readily, literally at the press of a button.

Furthermore, developments in robot technology do not benefit only those industries engaged in batch manufacturing. Robot technology can also help those more traditional mass-production-line industries. As the technology evolves, new small, fast robots are emerging from laboratories. These robots can keep pace with very fast conveyors, working on passing items with the speed and accuracy of a human.

Robots now are being used extensively in welding, die casting, materials handling, spray painting, loading and unloading machines, and inspection. Says Kovaly: "They will become increasingly common in assembly techniques as automated, programmable assembly systems take shape."

As visual, tactile and proximity sensors become integrated into robot systems, these articulated arms will manipulate tools with ease, he foresees. "With tools in hand, robots will be commonplace in finishing operations such as grinding and deburring and in arc welding."

Concludes Kovaly: "The time is at hand when some of industry's most unpleasant, demanding and demeaning jobs no longer need a human touch. This transition to automation will be explored every month by *Industrial Robots*."

For more information, write Technical Insights Inc., P. O. Box 1304, Fort Lee, NJ 07024. Circle 16

Monolithic 10-Bit, Tracking A/D Converter from Datel

Datel Intersil has come out with a new 10-bit, tracking analog-to-digital converter called the ADC-856, a versatile device with a wide range of applications.

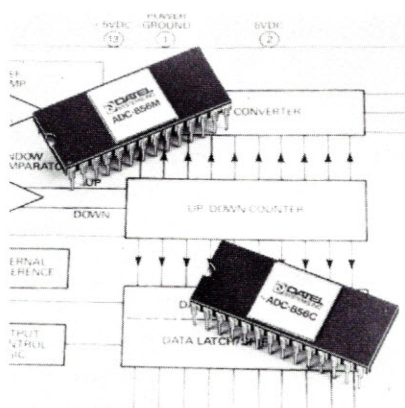
The ADC-856 can supply continuously updated conversion data about full-scale sinusoidal signals of up to 300 hertz without a sample and hold. The converter is linear to ± 0.5 LSB (min.) and is monotonic over its entire operating temperature range.

The circuit is implemented in bipolar, monolithic form. The chip contains a fast window comparator, tracking logic, an up/down counter, a digital-to-analog converter, a precision voltage reference with amplifier, data transfer gates and a data latch/shift register.

The external parts required for operation have been held to a few passsive components and allow external programming of the analog input voltage range.

The ADC-856 is optimized for operation in a continuous tracking mode. Each conversion of an analog signal is based on the last converted value of that signal. For signals that do not vary faster than the converter can track (1 LSB per microsecond), continuous tracking provides a valid, updated conversion result every microsecond.

Write or call for more information: Datel Intersil, 11 Cabot Blvd., Mansfield, MA 02048. 617/828-8000 or 339-9341. Circle 17



TECHNICAL ABSTRACTS

As a part of our goal of disseminating current technical information to our readers, this department will list abstracts of significant recent technical papers, in cases where these papers are available to the public. The relevant addresses will normally be listed after the abstracts. We urge academic and industrial research centers to send us abstracts of recent papers in Robotics and Artificial Intelligence for possible inclusion in this department, with appropriate prices and ordering procedures.

Experiments in Part Acquisition Using Robot Vision, by R. N. Nagel, G. J. Vanderbrug, J. S. Albus, and E. Lowenfield, National Bureau of Standards, SME MS79-784

The vision system for robots being investigated at NBS is mounted on the wrist of the robot, and provides both depth and part orientation information to the robot control system. The principle components of the vision system are a solid state camera, a structured light source, and a camera interface system. In experiments performed with the NBS vision system, the robot has been able to acquire both rectangular and curved parts. This paper reviews the hardware configuration, provides an overview of the software and describes the experiments.

The Remote Axis Admittance-A Key to Robot Assembly, by Paul C. Watson, Chief Engineer, Assembly Assoc., SME MS79-798

The Remote Axis Admittance (RAA) is a new device to perform the fine motion of parts mating. It can be mounted on the end of a robot arm, and will perform the final phases of assembly when the parts are in close proximity and when they are in contact. The RAA can be adapted to a number of different types of tasks, including the simple insertion, such as the chamfered peg and hole, the insertion of edges in slots, the multiple insertion, and the chamferless insertion. The RAA has optional built-in sensing and actuation, depending on the task, and uses a microcomputer type of controller to monitor and switch its different modes of operation. The RAA has built-in safety features and inherent internal damping.

Shape Segmentation Using Relaxation, by W. Rutkowski, S. Peleg, and A. Rosenfeld, TR-762

Relaxation is applied to the segmentation of closed boundary curves of shapes. The ambiguous segmentation of the boundary is represented by a directed graph structure whose nodes represent segments, where two nodes are joined by an arc if the segments are consecutive along the boundary. A probability vector is associated with each node; each component of this vector provides an estimate of the probability that the corresponding segment is a particular part of the object. Relaxation is used to eliminate impossible sequences of parts, or reduce the probabilities of unlikely ones. In experiments involving airplane shapes, this almost always results in a drastic simplification of the graph, with only good interpretations surviving. (30 pages, $1.50)

Texture Primitive Extraction Using an Edge-based Approach, T. Hong, C. Dyer and A. Rosenfeld, TR-763

Many textures are characterizable as a collection of primitive elements arranged over a background field. This paper defines an edge-based procedure for extracting primitives from textures. The technique groups edges into region boundaries by joining facing pairs of edge points. A pilot evaluation is performed by examining the usefulness of these primitives for texture classification. (33 pages, $1.65)

Ordering information for reprints of SME Technical Papers may be obtained from:
Society of Manufacturing Engineers
One SME Dr., PO Box 930
Dearborn, MI 48128
Phone 313/271-1500

U. of Md Technical Reports 762-3 may be ordered from:
Dr. Azriel Rosenfeld
Computer Vision Lab
University of Maryland
College Park, MD 20742

BOOKS

How to Build Your Own Self-Programming Robot, by David L. Heiserman (TAB Books #1241, Blue Ridge Summit, PA, 1979, 230 pages), is a successor to the author's *Build Your Own Robot* (TAB Books #841), which described the construction of a simple robot with little on-board logic. In this book, however, a robot with considerably more compute power is described, called "Rodney," which functions as an autonomous, adaptive machine.

The book begins by presenting the goals of the project and an explanation of the different classes of robot as defined by the author. By Heiserman's definition, a *true* robot must be self-programming, that is, exhibit behavior patterns that are not explicitly programmed by its builder. He defines an *Alpha*-class robot as one whose behavior is limited to basic reflexes, such as the robot described in his first book. In this case, the basic reflex is to randomly select a new direction of motion whenever the robot encounters an obstacle. The first stage of construction results in an Alpha-level "Rodney" based on the Intel 8085 microcomputer.

A *Beta*-class robot is slightly more "intelligent," having the ability to remember (in RAM) those stimulus-response (S-R) patterns that were "successful" in the past, and repeating the response upon receiving the same stimulus. In this case, success is measured in terms of avoiding obstacles or finding the robot's "feeding" (recharging) station. The addition of RAM and new programming raise Rodney to the Beta level.

Heiserman's definition of a *Gamma*-class robot is one that has the capabilities of the Beta class, but can also "generalize" the responses learned. Although his notion of generalization remains rather vaguely defined, the Gamma-Rodney programming does implement a form of generalization by transferring information from "successful" S-R patterns to "unsuccessful" ones based on preprogrammed relevance criteria. The addition of various sensors for light, sound, and a low-voltage condition allows Rodney to include new information in its stimulus patterns, permitting more varied behavior. A cassette interface allows Rodney's control programs to be saved and reloaded conveniently.

The book provides an introduction to the assembly and trouble-shooting procedures used in the robot's construction. Techniques are described that readers may find useful in other projects, such as a motor speed control circuit based on feedback from an LED and sensor in the wheel gearbox. Schematics and assembly language programs are supplied, but much of the circuit layout must be designed by the builder, which may prove a handicap to the inexperienced. There are detailed explanations for the circuits and programs, but very few photographs of the finished products—which might have served as a guide to the builder.

Due to the substantial scope of the project and the consequent volume of material to be explained, the book does not include any in-depth education on circuit design or the programming requirements for advanced robot behavior. It presumes experience with board layout and mechanical assembly, and some previous microcomputer programming experience is desired, though not essential for the highly motivated reader. Nonetheless, the book serves as a fair introduction to robot fundamentals and provides a complete explanation of the design and construction of Heiserman's robot.

Another problem with the book is that, although terms like "learning" and "generalization" are used quite freely, the concepts behind them are glossed over only superficially. Thus, the book relies heavily upon the ideas those terms *suggest* to the reader, which can easily result in misconceptions about the complexity of these concepts in the context of robotics and machine intelligence. Only the informed and observant reader will realize the vast difference between the behavior expressed by Heiserman's programs and that of higher animals, and the immense difficulty of bridging that gap.

How to Build Your Own Working Robot Pet, by Frank DaCosta (TAB Books #1141, 1979, 239 pages), is another in TAB's "How-to" robot construction series. The goal of this project is to construct a robot that simulates a dog, right down to the "bark" and the "wag." The format of the book is much like Heiserman's, beginning with a brief overview of the project, explaining the various capabilities desired and the systems designed to accomplish them.

Like Rodney, this project is based on the Intel 8085 MPU, and features a few unique circuits to give the robot an animal character. These include a sound generator that can simulate a bark or other animal noise, a primitive sound recognition unit that can distinguish between tones at frequencies above or below a variable reference tone (with a

program that can decode tone sequences into robot commands, and of course, a tail wagger. The robot also includes an ultrasonic ranging unit, a separate power supply, and a KC Standard cassette interface.

The author includes schematics and instructions for assembling all circuits and the robot chassis, as well as a short course on programming the 8085. A machine language program is included that will allow the robot to respond to commands, move around, and utilize the sonic range data, impact sensors, and bark and tail-wagger outputs. The program is listed in source mnemonics and hex code, although the listing format is non-standard. The program is called "ARASEM," for ARtificially Random SElf-Motivation, using the output of a random number generator to determine the robot's behavior.

Like Heiserman's book, all circuits, assembly and programming are explained in detail in a conversational style, but the project requires that the builder be adept at mechanical assembly, circuit layout design and wire-wrapping. These books are quite useful for those wishing to build the particular robot described, but what might be more desirable to many readers, instead of a continuing series of self-contained construction projects, is a description of various circuits and sensors and their application to robotics, with explanations that would allow the reader to generalize the designs to other motors, sensors, etc. This, with accompanying discussions of software subroutines that work in conjunction with the circuits, would allow the builder to configure his own robot from a set of "stock" elements, producing a unique creation instead of repeating another's design.

—*reviews by David Goodman*

ORGANIZATIONS

(continued from page 43)

industrial robotics and automated assembly operations, as well as the three new areas introduced at Autofact II: computerized quality inspection, material handling and predictive maintenance.

Exhibit details are available now for Autofact West and will be available this fall for Autofact III, Hilty said.

For more information, write or call Bill Hilty, Society of Manufacturing Engineers, 1 SME Drive, P. O. Box 930, Dearborn, MI 48128. (313)271-1500, ext. 300.

The Robot Institute of America

The Robot Insitute of America recently asked itself a rather basic but important question: Just what is a robot?

The answer, now the formal RIA definition of robot, was: "a programmable, multifunction manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks."

"The need for a definition was obvious," said RIA President Jerry Kirsch. "The acceptance of robotic technology can be seen in many industries, and the public's awareness of robotics has increased. This definition clearly states the key attributes of devices commercially available in the United States."

The definition was one of several considered by an RIA committee of robot users, manufacturers and researchers.

RIA was founded in 1974 as a trade association to serve worldwide robot technology. It claims as members all major US robot manufacturers.

The institute, in cooperation with the Society of Manufacturing Engineers, serves industry through a variety of educational activities: seminars, conferences, expositions, research reports, case studies, films and publications, including *ROBOTICS TODAY*, a quarterly trade journal on industrial robotics.

For more information about RIA, write or call its manager, Donald A. Vincent, Robot Institute of America, 1 SME Drive, P. O. Box 930, Dearborn, MI 48128. (313)271-1500, ext. 404.

CLASSIFIED

Our classified advertising section is for readers wishing to buy, sell or trade hardware or software. The price is 30¢ per word. The first word is set in all caps. Minimum 20 words. Send copy with check to ROBOTICS AGE CLASSIFIED, P. O. Box 801, La Canada, CA 91011.

MANUFACTURERS: Receive a FREE listing in the forthcoming ROBOTICS PRODUCT INDEX in ROBOTICS AGE Magazine. Here is a partial list of product categories: Motors; Power Trains; Computers and Interfaces; TV Input; Speech Input; Subassemblies; and more. Send your entries to: ROBOTICS PRODUCTS INDEX, ROBOTICS AGE Magazine, P. O. Box 801, La Canada, CA 91011.

LOS ANGELES AREA—anyone interested in joining the first local chapter of the International Robotics Foundation with regular meetings and activities should call 213/790-5823 and leave your name, address and phone number.

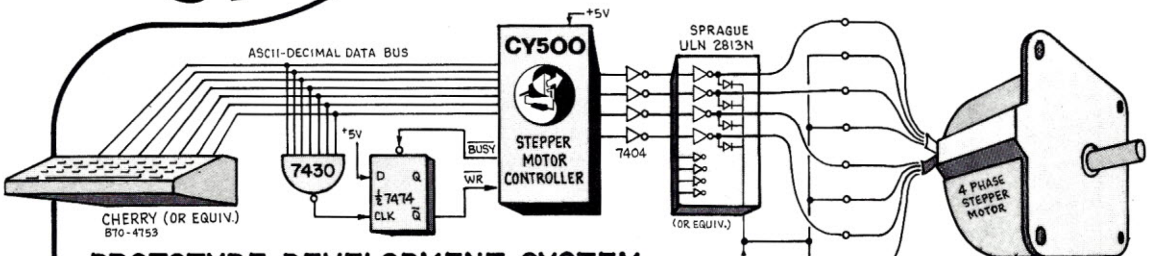
"A How-To-Do-It Manual on Non Court-Ordered Wiretapping..." That's what one congressman called a remarkable government-funded study — "Selected Examples of Possible Approaches to Electronic Communication Interception Operations." This study, which we have obtained under the Freedom of Information Act, gives step by step instructions on how to tap into: a residential phone; a trunk phone line; individual conversations on microwave relays between cities; even interception of a business's computer data transmissions is covered. Only by knowing what is possible can you hope to protect your personal and business communications from interception. This study is a must for all who value their privacy. Available for $4.50 from: Microbus, P.O. Box 4029, Houston, TX 77210.

CY500

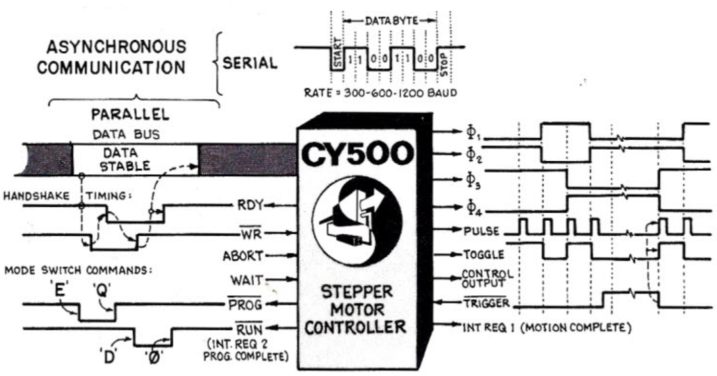
STORED PROGRAM STEPPER MOTOR CONTROLLER

CYBERNETIC • MICRO SYSTEMS •

THE CY500 STEPPER MOTOR CONTROLLER UTILIZES A HI LEVEL LANGUAGE FOR CONVENIENT CONTROL OF DIRECTION, POSITION, SPEED, AND ACCELERATION OF ANY FOUR PHASE STEPPER MOTOR. THE SELECTION OF EITHER BINARY CODED OR ASCII-DECIMAL CODED COMMANDS ALLOWS THE USE OF A SIMPLE ASCII KEYBOARD FOR PROTOTYPE DEVELOPMENT IN SIMPLE SYSTEMS.



PROTOTYPE DEVELOPMENT SYSTEM



CY500 TIMING & CONTROL SIGNALS

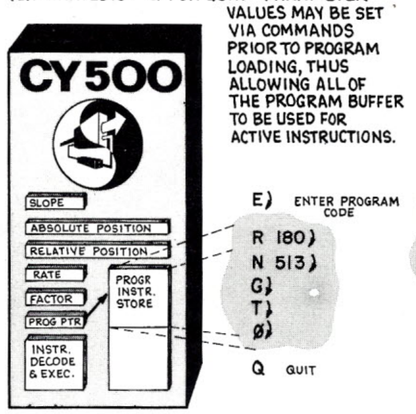
CY500 COMMANDS

- **A** ATHOME (DECLARE Ø POSITION)
- **B** BITSET (CONTROL OUTPUT = 1)
- **C** CLEARBIT (SET CONTROL = Ø)
- **D** DO IT NOW (EXECUTE PROGRAM)
- **E** ENTER (PROGRAM INTO CY500)
- **F**f FACTOR (DIVIDES RATE BY f)
- **G** GO (BEGIN STEPPING)
- **H** HALFSTEP MODE
- **I** INITIALIZE CY500
- **J** JOG (EXTERNAL START/STOP)
- **L** LEFT/RIGHT (EXT. DIR. CONTROL)
- **N**n NUMBER OF STEPS n
- **O** ONESTEP (IMMEDIATELY)
- **P**p POSITION p IS DESTINATION
- **Q*** QUIT PROGRAM MODE
- **R**r RATE OF STEPPING SET TO r
- **S**s SLOPE OF ACCELERATION (±s)
- **T** TIL PIN 26 HI, REPEAT PROGRAM
- **U** UNTIL 'WAIT' LOW, WAIT HERE
- **+** SET CLOCKWISE DIRECTION
- **-** SET COUNTERCLOCKWISE DIR.
- **Ø** RETURN TO COMMAND MODE

*NO CARRIAGE RETURN AFTER Q

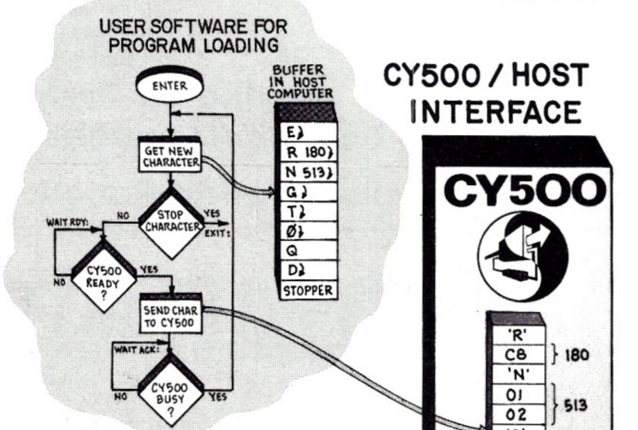
CY500 PROGRAMMING EXAMPLE

CONSIDER AN INSTRUCTION SEQUENCE TO BE STORED AS A PROGRAM THAT WILL CAUSE THE CY500 CONTROLLER TO STEP A MOTOR FOR 513 STEPS WITH A RATE PARAMETER OF 180. AFTER COMPLETING THE TRAVEL, THE PROGRAM SHOULD TEST AN EXTERNAL CONTROL LINE AND EITHER REPEAT THIS BEHAVIOR OR RETURN TO THE COMMAND MODE. TO ENTER THIS COMMAND SEQUENCE INTO THE CY500 PROGRAM BUFFER WE SEND 'E' FOLLOWED BY '}' (=ØDH) THEN THE COMMAND STRING, TERMINATED BY 'Q' FOR QUIT. PARAMETER VALUES MAY BE SET VIA COMMANDS PRIOR TO PROGRAM LOADING, THUS ALLOWING ALL OF THE PROGRAM BUFFER TO BE USED FOR ACTIVE INSTRUCTIONS.



WE BEGIN EXECUTING THE STORED PROGRAM BY SENDING 'D' (FOR DO IT NOW). THE 'D' COMMAND MAY BE EMBEDDED IN A PROGRAM TO CAUSE REPEATED EXECUTION (LOOPING) OF A PROGRAM.

CY500 / HOST INTERFACE



HOST SOFTWARE CONSISTS OF BUFFER TO HOLD COMMANDS TO BE LOADED INTO CY500 PROGRAM BUFFER PLUS HAND SHAKING ALGORITHM TO COMMUNICATE WITH CY500.

THE CY500 STORED PROGRAM STEPPER MOTOR CONTROLLER IS A STANDARD 5-VOLT 40-PIN LSI DEVICE CONFIGURED TO CONTROL ANY 4-PHASE STEPPER MOTOR. THE CY500 IS AVAILABLE FROM STOCK AT $95 SINGLE QUANTITY.

Cybernetic Micro Systems
445-203 So. San Antonio Road • Los Altos CA 94022
Phone: 415-949-0666 • Telex: 171-135 attn: Cybernetic